Special Programming Issue

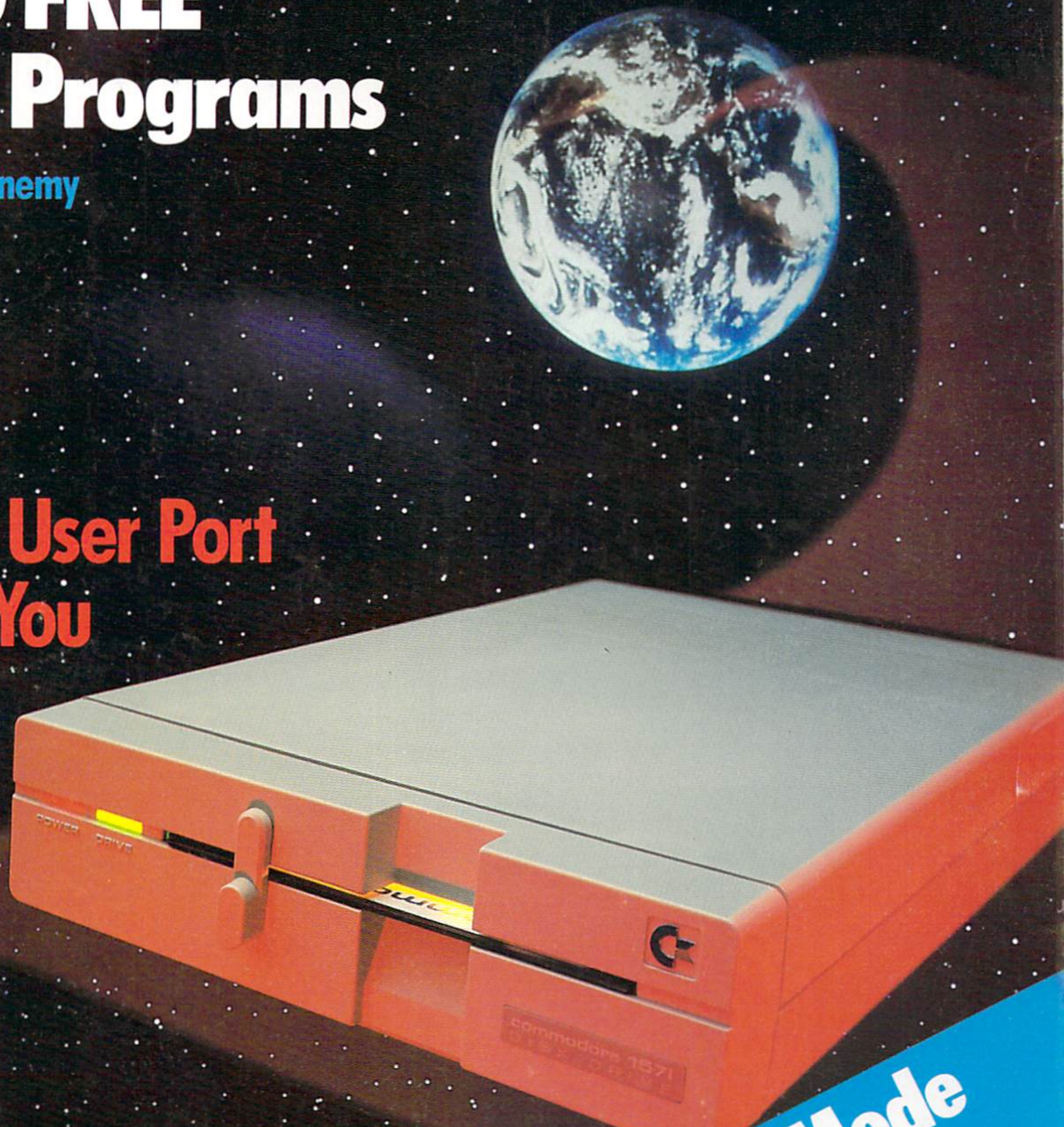
April 1987
$2.95 U.S.
$3.95 Canada
ISSN 0744-8724

commodore MAGAZINE

Over 20 FREE Type-in Programs

Falcon and the Enemy
SDI Star Wars
Moonlight Zone
AmigaLife
Inter Melody
And Many More

Make the User Port Work for You



Burst Mode on the 128

We just did something only the best can do! We made our award winning* software for the Commodore™ 128 and 64 computers even better!

Introducing . . .

Pocket Writer 2
word processor
Pocket Planner 2
spreadsheet
Pocket Filer 2
database

New Features

Our new Pocket 2 series offers features usually found only in much more sophisticated applications software. Features that include: compatability with the new GEOS operating system†, ability to work with the Commodore RAM expander to allow a RAM disk, mouse support with pull down menus, 1571 burst mode for faster file loading, increased support for two single disk drives, automatic configuration for screen color, format and printer selection†.

Sophisticated software, yes, and still easy to use. You can be up and running in under 30 minutes even if you haven't operated a computer before.

2 Programs in 1

Now, when you upgrade your Commodore™ 64 to a 128, Pocket software helps make it a breeze. The new Pocket 2 software has both 128 and 64 applications on the same disk. So when you buy one you are actually buying two software packages. The cost only $59.95 (U.S.).

6 Programs in 1

The 180% Solution saves you money! You can buy all three Pocket 2 applications, Pocket Writer 2, Pocket Planner 2 and Pocket Filer 2 in one convenient **Superpak** for the low price of only $99.95 (U.S.). A super way to discover all the integrated features of Pocket 2 software and save almost eighty dollars.

As a companion to Pocket Writer 2, a **Dictionary Disk** containing 32,000 words (expandable to 40,000) is available. The cost $14.95 (U.S.).

For those of you who have already discovered the many benefits of owning Pocket software; we offer all registered owners an upgrade to Pocket 2 software for only $19.95 (U.S.) plus 3.00 (U.S.) shipping and handling! Available only by writing to Digital Solutions Inc.

Pocket Writer 2 Word Processor
In addition to the new features above . . .
- Spelling Checker incorporated in program (requires a dictionary disk)
- Spelling Checker now runs over 300% faster than in original Pocket software
- Word wrap is now fully automatic†
- Ability to move columns
- Go To page number for finding information in long texts†
- Fully automatic upper and lower case type conversion†
- Enhanced Delete process for word, line or paragraph
- Word Count feature for essays and assignments†
- Enhanced split memory mail merge option

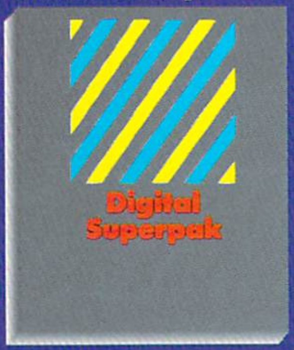
Pocket Planner 2 Spreadsheet
In additon to the new features above . . .
- Individual column width selection now available†
- Multiple files in memory with cut and paste capability
- Able to print mathematical formulae as well as results of calculations†
- Global formatting option
- Enhanced row/column insert delete†
- Logarithmic and XY graphing capability
- Increased file compatability with other spreadsheets†
- Number of rows increased from 99 to 250†

Pocket Filer 2 Database
In addition to the new features above . . .
- Dynamic calculations during data entry
- Intelligent re entry to enter/edit mode
- Easier file conversion from other software†
- Automatic index updating for constantly sorted file†
- Enhanced mathematical language including loops and labels†
- High speed sort using dynamic buffering†
- Automatic entry of repetative data†

Serious Software That's Simple to Use

*Commodore's Microcomputers Magazine, independent reviewers, rated the original Pocket Writer 128/64 and Pocket Planner 128/64 software the "Annual Best of 1986" in the productivity category.

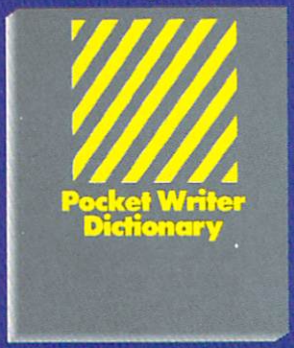
Superpak: The Solution That Saves Money!



- Pocket Writer 2, Pocket Planner 2 and Pocket Filer 2 together
- Convenient; get all three integrated applications at once
- 128/64 software on same disks
- Economical; $179.85 (U.S.) worth of software for only

$99.95 (U.S.)

Pocket Writer Dictionary



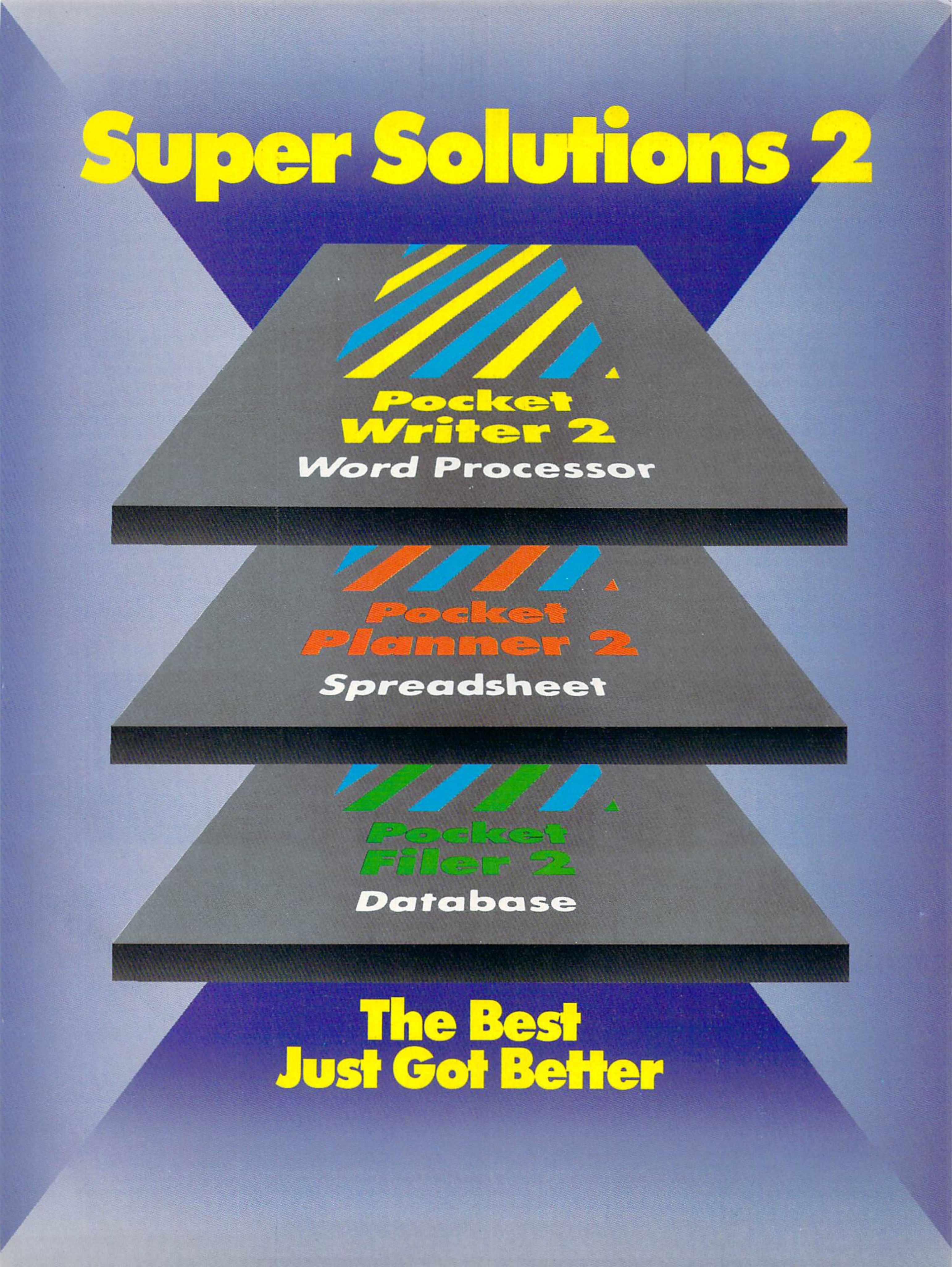
- Makes Spelling Checker faster and simpler to use
- More convenient than developing personal disk
- 32,000 words available
- Expandable to 40,000 words

Canadian Dealer Enquiries:
Ingram Canada Ltd.
1-416-738-1700

Mail orders:
Crystal Computer Inc.
in Michigan 1-517-224-7667
outside Michigan 1-800-245-7316

International Distributor Enquiries to:

Digital Solutions Inc.
2-30 Wertheim Court
Richmond Hill, Ontario
Canada L4B 1B9
Telephone (416) 731-8775
Telex 06-964501
Fax (416) 731-8915


Super Solutions 2
Pocket
Writer 2
Word Processor
Pocket
Planner 2
Spreadsheet
Pocket
Filer 2
Database
The Best
Just Got Better

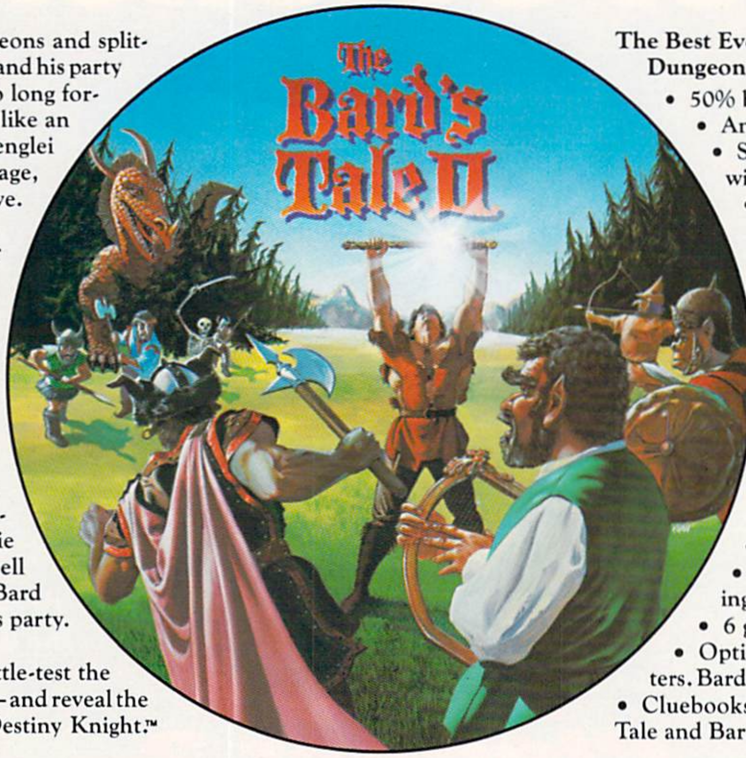
"The Bard is Back!"

From impossible dungeons and split-second snares, the Bard and his party emerge. The Sceptre, so long forgotten, gleams with power like an exploding sun. Even Phenglei Kai, the ancient archmage, bows his head in awe.

"I smell serpents!" Slipfinger squeals, stealing away like the thief he is. Two arch-dragons slither out of the ground, their eyes burning with the relentless fury of treasure lost.

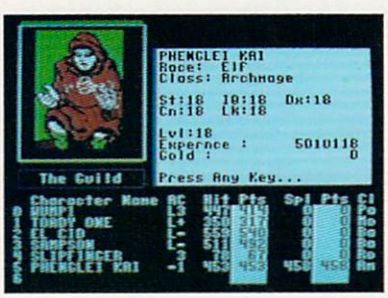
Protected behind the flame lizards, beyond the reach of normal weapons, a cackling wizard begins the eerie chants of a death spell. A spell that can finish the Bard and his party.

The time has come to battle-test the magic of the Destiny Wand – and reveal the awesome powers of The Destiny Knight.™



The Best Ever Dungeon Role-Playing Game

- 50% bigger than Bard's Tale.™
- An all-new story line.
- Six cities and a huge overland wilderness to explore.
- Dozens of new spells – 79 spells in all.
- New real-time dungeon puzzles. You have to get through them before the clock stops ticking.
- Summon and name monsters to become a permanent part of your party.
- More strategy in combat encounters – the weapons and spells you choose depend on the enemy's distance.
- A bank and casino.
- A starter-dungeon for building up your low-level characters.
- 6 guilds for easier game saving.
- Optional use of Bard's Tale characters. Bard's Tale experience not required.
- Cluebooks available for both Bard's Tale and Bard's Tale II.™



You get a new class of magic user – the Archmage. With 8 powerful spells like Heal All, Fanskar's Night Lance, and the awesome Mangar's Mallot.



There are over 100 monsters, like this Kner Drone. Many animated. All dangerous.



25 scrolling dungeon levels. All in color. All 3-D. Including 7 different Snares of Death, a new kind of real-time puzzle.

The Bard's Tale II

The Destiny Knight

Optional use of characters from Ultima III™ or Bard's Tale™

from

ELECTRONIC ARTS™

HOW TO GET IT: Visit your retailer, or call 800-245-4525 (in CA call 800-562-1112) for VISA or Mastercard orders. To buy by mail, send a check, money order, or VISA or Mastercard information to Electronic Arts, P.O. Box 7530, San Mateo, CA 94403. The price is $39.95 for the Commodore 64 version. Add $5 for shipping and handling ($7 Canadian). Allow 1-4 weeks for delivery. The Bard's Tale II and Electronic Arts are registered trademarks of Electronic Arts. Ultima is a registered trademark of Richard Garriott. Commodore is a trademark of Commodore Electronics Ltd. For a copy of our complete catalog, send 50¢ and a stamped, self-addressed envelope to Electronic Arts Catalog, 1820 Gateway Drive, San Mateo, CA 94404.

VOLUME 8, NUMBER 4

CONTENTS

APRIL 1987

commodore MAGAZINE

DEPARTMENTS

FEATURES

LETTERS

How to Build a Light Pen

To the Editor:

I would like to thank you for publishing "How to Build a Light Pen" in the January 1987 issue of *Commodore Magazine.* As a result of the article, I've already constructed several of the light pens for my friends and for myself.

I would like to point out that the Motorola Type MRD 300 phototransistor is quite hard to locate. However, they are available from:

Pioneer/Washington
9100 Gaither Road
Gaithersburg, MD 20877
Tel: (301) 921-0660

The cost for each of the MRD 300's is $1.88 each and a $10.00 minimum order is required.

I checked all local distributors in an attempt to locate the MRD 300 and finally, through Motorola Customer Assistance, was able to locate them. I would also like to point out that the MRD 300 is far superior to the alternate Radio Shack phototransistor.

Just one word of caution concerning assembly of the light pens. Take your time in assembling them and try to locate a magic marker slightly larger than what is indicated in the article.

I am currently using the light pens with Commodore's *Micro Illustrator* with little problems, and I am experimenting with some other software packages.

Keep up the good work.

John Orberson
Frankfort, Kentucky

What to Do with Your Christmas $$

To the Editor:

In the article about what to do with your Christmas money in the November/December 1986 issue, I took offense at your comment on the Coleco Adam computer in the paragraph about the chocolate computer sold by Long Grove Confectionery Company (". . . and runs just as much software as a Coleco Adam. Maybe more.")

I have both a Commodore 64 and an Adam. While I use the 64 for serious work, I use the Adam for playing games — and very good games too! Great games that were never released for the 64 like *Slither, Turbo, Dukes of Hazzard, Subrock, Space Panic, Carnival, Smurfs, Bump 'n Jump, Bergertime, Venture, Ladybug, Frenzy, Time Pilot,* and *Mousetrap.* And my personal favorite, *Cabbage Patch Kids* — there is nothing like the sight of seeing a crying Cabbage Patch Kid sink in a tar pit!

Mr. Dennis R. Schliebener
Louisville, Kentucky

Copy Protection

To the Editor:

Al Metz stated in his letter in the February issue that most companies will sell you a replacement copy for a minimal fee, and that you shouldn't even need that since your software should last indefinitely, barring any accidents.

I don't agree with this. Let's say you do have an accident. First, what if you can't afford to wait for a copy to arrive from the company because you're working on a crucial project?

Or what do you do if the software is licensed and the creators of the program can no longer sell the program or a backup copy of it? This happened with Microsoft's *Multiplan,* which was licensed to Epyx.

Or what happens when a company has brought out a newer version of a program and will not sell you a backup because you own an older version of the program. This happened to me with *VIP Professional,* version 7.1.

Both these programs are very good, and I do not regret buying them. But they do illustrate that sometimes you cannot get backup copies of software from the manufacturer.

Daniel Lewis
Butte, Montana

Don't Panic—Yet

To the Editor:

I would like to congratulate and say thank you to Gary V. Fields. His article in the February issue of *Commodore Magazine* is very good and to the point. I am a new user and many times I have blamed the equipment and software, but so far it has always been my fault.

I have had that feeling of panic and probably will many more times. It is so easy to forget one thing that can cause you all kinds of problems. But as he says, if you stop and check out what you are doing step by step, it sometimes helps.

I just had to comment on this article because I enjoyed chuckling to myself at the truth of it.

Robert E. Decker
Glenn, Michigan

STAFF

Publisher
Diane C. LeBold
Assistant to the Publisher
Kelly McKeown

Editor
Carol Minton
Technical Editor
Jim Gracely
West Coast Correspondent
Matthew Leeds

Art Director
Gwenn Knapp
Assistant Art Director
Wilson Harp
Production Assistant
Bob Clark
Cover Photo
Gene Smith/NASA
Production Manager
Jo-Ellen Temple

Circulation
Kenneth F. Battista
Advertising Coordinator
Rebecca Cotton

Advertising Representatives
SOUTHEAST, SOUTHWEST AND WEST COAST
Warren Langer, Spencer O. Smith
Warren Langer Associates
9320 NW 2nd Street
Coral Springs, FL 33071
Advertising Inquiries Only
305/753-4124
MIDWEST, NORTHEAST AND CANADA
Pamela Stockham
700 River Road
Fair Haven, NJ 07701
201/741-5784

Commodore Magazine, Volume 8, Number 4, April 1987, ISBN 0-88731-068-0.

Commodore Magazine (ISSN 0744-8724) is published monthly by Commodore Magazine Inc., 1200 Wilson Drive, West Chester, PA 19380. U.S.A. U.S. subscriber rate is $35.40 per year; Canadian subscriber rate is $45.40 per year; Overseas subscriber rate is $65.00 per year. Questions concerning subscription should be directed to Commodore Magazine Subscription Department, Box 651, Holmes, Pennsylvania 19043. Phone (800) 345-8112. In Pennsylvania (800) 662-2444.

ABC Membership applied for.

tcp DOES MAIL ORDER RIGHT



GREAT PRICES
"We buy in volume and pass the savings onto you!!!"

FAST SERVICE
"Over 90% of all items ordered are in stock and shipped within 24 hours."

FEDERAL EXPRESS
FEDERAL EXPRESS

"Software orders over $50.00 will be shipped Federal Express. You only pay TCP's standard shipping charge of $4.00 per order. This offer also valid on peripherals and accessories under 8 pounds. Orders arriving before 11:00 am our time will be shipped out same day."

BEST SELECTION
"We stock hundreds and hundreds of products for commodore computers! Every week we add even more. We're the first to get in all the hot new software releases. Just another reason why we should be your first call!!"

"I'm proud we deliver all of these. Response from tens of thousands of satisfied customers can only mean we're doing mail order right!"

Peter Sattler President
TUSSEY COMPUTER PRODUCTS

EDUCATIONAL & CORPORATE ACCOUNTS:

We offer a full line of services catering to you. **Including; quick turn-around, personal service, and the flexibility to serve your organizations needs. Credit Approval in 24 hrs for any organization with a good D&B rating.**

To reach our Educational and Corporate Sales Department directly, Call 1-800-533-1131 or Inside PA Call 814-234-2236

OTHER CUSTOMERS PLEASE CALL THE NUMBER LISTED BELOW

FREE COMMODORE 1660 MODEM COUPON
with the purchase of any of these packages ordered MAR 29th or before*



Commodore 128 computer 1571 Disk Drive	$478.00
Commodore 128 computer 1571 Disk Drive 1902A Monitor	$737.00
Commodore 64C computer 1541 Disk Drive	$349.95
Commodore 64C computer 1541 Disk Drive 1802C Monitor	$549.95

*Coupon is redeemable by mail for (1) 1660 Modem - Call for more information.

CALL TOLL FREE 1-800-468-9044 Inside PA Call 814-234-2236

tcp DOES MAIL ORDER RIGHT

OPEN: 9am-8pm Mon-Fri, 10am-5pm Sat EAST COAST TIME
NEW HRS STARTING MAR 01: 9am-9pm M-Thur, 9am-6pm Fri, 10am-6pm Sat, Noon-6pm Sun

INFORMATION AND PA ORDERS 814-234-2236

GAME GALLERY Educational Software too! All Games stocked for quick ship!!!

NEW TITLES IN STOCK EVERYDAY!! If you don't see what you want please call!!



MISC GAMES
Hobbitt	$CALL
Sargon III	$37.95
Murder by the Dozen	$24.95

ACCESS
Beach Head	$24.95
Beach Head 2	$29.95
Leader Board	$29.95
Leader Board Tourn. Disk	$16.95
Tenth Frame	$27.95

ACCOLADE
Dam Busters	$22.95
Fight Night	$22.95
Hard Ball	$22.95
Law of the West	$22.95
PSI 5 trading co	$22.95
Spy vs Spy	$12.49

ACTIONSOFT
Call for price on all Actionsoft titles

ACTIVISION
Alcazar	$19.95
Borrowed Time	$19.95
Comp. Firewks Celebrat'n	$19.95
Countdown to shutdown	$19.95
Fast Tracks Slot Car Const	$19.95
Gamemaker Scifi Library	$15.95
Gamemaker Sports Library	$15.95
Garry Kitchen's Gamemakr	$24.95
GBA Basketball 2 on 2	$24.95
Ghostbusters	$24.95
Great Amer RR	$24.95
Hacker	$19.95
Hacker II	$24.95
I am the 64	$21.95
I am the 128	$24.95
Little Computer People	$24.95
Master Of Lamps	$19.95
Mindshadow	$19.95
On Court Tennis	$21.95
Pitfall II: Lost Caverns	$19.95
Shanghai	$26.95
Space Shuttle	$19.95
Star Rank Boxing	$21.95

ARTWORX
International Hockey	$18.95

AVALON HILL
Dr Ruth	$21.95
Gulfstrike	$22.95
Spitfire 40	$24.95
Superbowl Sunday	$26.95
Team Disk for SS	$16.95

BRODERBUND
Champshp Lode Runner	$24.95
Karateka	$19.95
Lode Runner	$24.95
Music Shop	$29.95

BLUE CHIP
Baron	$24.95
Millionaire	$24.95
Tycoon	$24.95

COMMODORE
Sky Travel	$27.95

DAVIDSON & ASSOR
now in stock! Call for price on titles

DATA EAST
Commando	$24.95
Karate Champ	$26.95
Kung Fu master	$26.95

DATA SOFT
Pole Position	$15.95

ELECTRONIC ARTS
Adventure Constr	$27.95
America's Cup	$24.95
Amnesia	$29.95
Archon	$12.95
Archon 2	$27.95
Artic Fox	$24.95
Autoduel	$37.95
Bard's Tale	$32.95
Bard's Tale II	$29.95
Battlefront	$29.95
Carriers at War	$37.95
Chessmaster 2000	$26.95
Cut and Paste	$12.95
Europe Ablaze	$37.95
Hard Hat Mack	$13.95
Heart of Africa	$12.95
LordS of Conquest	$24.95
Mail Order Monsters	$12.95
Marble Madness	$27.95
Moebius	$29.95
Movie Maker	$27.95
M.U.L.E.	$12.95
Murder Party	$25.95
Music Construction Set	$12.95
Ogre	$29.95
One on One	$12.95
Pinball Construction Set	$12.95
Quizam	$20.95
Racing Destruction Set	$12.95
Reach for the Stars	$32.95
Realm of Impossible	$12.95
Robot Rascals	$29.95
Seven Cities of Gold	$12.95
Sky Fox	$24.95
Software Golden Oldies	$16.95
Starfleet I	$32.95
Super Bolder Dash	$12.95
Tim Leary's Mind mirror	$24.95
Touchdown Football	$22.95
Ultima I	$29.95
Ultima III	$37.95
Ultima IV	$45.95

EPYX
Championship Wrestling	$27.95
Movie Monster	$22.95
Summer Games	$26.95
Summer Games II	$26.95
Super Cycle	$29.95
Temple Trilogy	$29.95
Winter Games	$27.95
World Games	$29.95
World's Greatest Baseball	$24.95
World's Greatest Football	$28.95
World Karate Cha	$22.95

Call for prices on other EPYX products !

FIREBIRD
Elite	$22.95
Pawn	$27.95

INFOCOM
Balley Hoo	$29.95
Cut Throats	$22.95
Deadline	$25.95
Enchanter	$25.95
Hitchhiker's Guide	$22.95
Infidel	$29.95
Leather Goddesses	$24.95
Moon Mist	$24.95
Planetfall	$25.95
Sorcerer	$29.95
Spellbreaker	$29.95
Suspect	$29.95
Trinity	$32.95
Wishbringer	$25.95
Witness	$25.95
Zork I	$22.95
Zork II	$26.95
Zork III	$26.95

INVISICLUE BOOKS FOR ANY INFOCOM GAME $6.95

LANCE HAFNER
Final Four Basketball	$29.95
Basketball, the Pro Game	$29.95

MICROLEAGUE
Microleague Baseball	$27.95
Microleague general mgr	$27.95
Microleague 1985 teams	$15.95

MICROPROSE
Acrojet	$24.95
Crusade in Europe	$27.95
Decision in the Desert	$27.95
F-15 Strike Eagle	$21.95
Gunship	
Conflict in Vietnam	
Hellcat Ace	$21.95
Kennedy Approach	$24.95
NATO Commander	$24.95
Silent Service	$24.95
Solo Flight	$24.95
Spitfire Ace	$21.95
Top Gunmen	

MINDSCAPE
Bank Street Music Writer	$27.95
Bank Street Storybook	$27.95
Bop-n-Wrestle	$21.95
Castle Clobber	$18.95
Halley Project	$27.95
Indiana Jones	$22.95
Indoor Sports	$22.95
Infiltrator	$21.95
Quake minus one	$17.95
Shadowfire	$17.95
Star Trek Promethian Adv	$29.95
Perfect Score:SAT prep	$49.95
The Lords of Midnight	$17.95

Call for prices on other MINDSCAPE products!

RADARSOFT
Floyd the Droid	$19.95
Maps USA	$34.95
Maps Europe	$34.95
World Maps	$34.95

SIERRA
Championship Boxing	$17.95

SIMON & SHUSTER
Paper Airplane const.	$CALL
JK Lasser's Income Tax	$39.95
Kermit's Story Maker	$19.95
NY Times Crossword Puzzle Vol. 1 or 2	$14.95
Spy Hunter	$31.95
Star Trek-Kobayashi alt.	$29.95
Typing Tutor III	$29.95

SPECTRUM HOLOBYTE
Gato	$CALL

SPRINGBOARD
Early Games	$26.95
Easy As	$29.95
Piece of Cake Math	$26.95

SUBLOGIC
Flight Simulator II	$32.95
Football	$37.95
FS II Scenery disk	$15.95
Jet	$29.95
Night Mission Pinball	$CALL
Pure Stat Baseball	$37.95

WEEKLY READER STICKY BEAR SOFTWARE
now in stock! Call for price on titles!

Random House, Spinnaker, and SSI products in stock!!!
Call for Price!

C-64 BUSINESS AND PRODUCTIVITY

C-64 DATABASES
Bank Street Filer	$34.95
Consultant	$39.95
Data Manager	$19.95
Fleet Filer	$29.95
Pocket Filer 64	$24.00
Profile 64	$36.95

C-64 INTEGRATED PKGS
Homepak	$39.95
Trio	$CALL
Vizastar 8k	$79.95

C-64 SPREADSHEETS
Calkit	$39.95
Pocket Planner 64	$CALL
Practicalc(d) or (t)	$CALL
PS, Programble Sprdsht	$19.95
Swiftcalc 64 w/sideways	$39.95
Sidways	$19.95
Vizastar 64	$79.95

C-64 WORD PROCESSORS
Bank Street Writer	$CALL
Bank Street Speller	$34.95
Cut & Paste(EOA)	$12.95
Fleet System II	$CALL
Font Master II	$34.95
Kid Pro Quo	$32.95
Paperclip	$37.95
Paperclip w/spellpack	$49.95
Pocket Writer	$CALL
Pocket Writer Dictionary	$19.95
Spellpro 64	$32.95
Trio	$CALL
Wordpro 3+/64	$14.95
Wordpro 64	$36.95
Wordpro GTS	$CALL
Word Writer 64 w/speller	$34.95

FINANCIAL & ACCT.
Cashbox	$36.95
Cont. Home Acct.	$46.95
Financial Time Machine	$29.95
Softsync Pers. Acct.	$32.95
Timeworks Electr. Checkbk	$19.95
Timeworks General Ledger, A/R,A/P, Payroll, Invtory ea	$40.95
Timeworks Money Mgr	$19.95

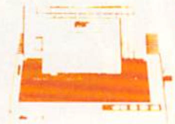
GRAPHICS
CADPIC	$32.95
Clip Art I	$19.95
Clip Art II	$CALL
Graphics Library I, II, or III	$16.95
Graphics Scrapbk 1 or 2	$17.95
Newsroom	$34.95
Picasso's Revenge w/pen	$CALL
Print Shop	$25.95
Print Shop Companion	$24.95

MISC. HARDWARE
Estes pwr supply for C-64	$54.95
Naverone 3 Slot expander	$27.95

UTILITIES
GT4	$22.95
Copy II 64	$24.00
C Power	$69.95
CSM 1541 align	$34.95
Fast Load	$24.95
Mach 5	$24.95
Merlin 64	$34.95
Pal 64	$32.95
Power 64	$32.95
Toolbox 64	$59.95
Quest Stat manager	$34.95
Vorpal fast loader	$CALL

FREE LIGHT PEN w/purchase of Picasso's Revenge Painting Program $42.95

PRINTERS

star


NX-10 PRICE DROP TO **$207.95**

NP-10	$184.95
NX-10C	$CALL
NX-15	$349.00
NL-10C	$CALL
POWERTYPE	$229.00

18cps, daisywheel

BROTHER 1509
180 CPS DOT MATRIX PRINTER, 15" CARRIAGE, FRICTION AND TRACTOR FEED, 45 CPS NLO MODE, & 3K BUFFER. USES EPSON FX PRINT CODES.



$379 OUR PRICE

EPSON
Epson Printers 1 yr warranty

LX-80	$229.00
HOMEWRITER 10	$CALL
FX-85e	$359.00
FX-286e	$CALL
RX-100	$279.00
DX-10	$CALL
HS-80 INKJET	$CALL

xetec
Fontmaster 128...	$CALL
Fontmaster 64	$34.95

SEIKOSHA
2 YR WARRANTY

SP-1000VC	$164.95

JUKI
6100P	$CALL
5510P includes color kit	$CALL

Printer Interfaces

MW-350 w/10k buffer	$CALL
XETEC Supergraphix	$CALL
XETEC Supergraphix Jr	$46.95

PANASONIC
2 YR WARRANTY

1080i	$199.95
1090i, 1092i	$CALL
3131 Daisywheel	$CALL



tcp TUSSEY COMPUTER PRODUCTS P.O. BOX 1006 STATE COLLEGE, PA 16804

Tips & Tricks

Hints for Fun and Utility

COMPILED BY LOUIS F. SANDER

Computing is a wonderful activity on a chilly day, and this month's trickery is sure to make your computing time more enjoyable. If you're into math at all, several of our tips are sure to interest you. If you're not into math, give them a try anyway. Another group of tips covers the nuts and bolts of connecting external devices to your computer. But as always, of course, there's a little something here for everyone.

If you have an idea, tip or trick that you'd like to share with others around the world, write it up and send it in. We pay from $10 to $50 for each item we use. Send your tricks to:

Louis F. Sander
P.O. Box 101011
Pittsburgh, PA 15237

If you include a self-addressed stamped envelope, we'll send you our hint-writer's guide.

Permutations and Combinations: These make up an interesting mathematical subject, readily amenable to computerization. Permutations and combinations are studied in college and in advanced high school math. Understanding them helps solve many problems in the area of probability and statistics.

A permutation is an ordered arrangement of a number of symbols or objects. XYZ, ZYX and YXZ are permutations of the sumbols X, Y and Z. A combination is a permutation that includes the same objects, regardless of the order in which they are arranged. XYZ, ZYX and YXZ are all different arrangements of the same combination. WXY, WXZ and WYZ are examples of different combinations.

If you wanted to know how many different batting orders would be possible on a nine-man baseball team, you'd ask a mathematician, "How many permutations are there of nine objects taken nine at a time?" If you could choose your batting order from a 24-man roster, you'd ask, "How many permutations are there of 24 objects taken nine at a time?"

The accompanying simple program can stand in for your mathematician, and can give answers a lot faster than he can. Just enter the number of objects (or numbers, players, whatever), and the number to take at a time. The answer will come up in milliseconds.

A. W. Grym
London, England

Permutations & Combinations

```
100 PRINT"[CLEAR,DOWN,SPACE2]
    PERMUTATIONS & COMBINATIONS -
    SANDER"
110 PRINT"[DOWN,SPACE2]
    THIS SHOWS THE NUMBER OF POSSIBLE"
120 PRINT"[SPACE2]PERMUTATIONS AND
    COMBINATIONS OF"
130 PRINT"[SPACE2]NUMBERS THAT YOU
    INPUT."
140 INPUT"[DOWN2,SPACE8]
    HOW MANY OBJECTS";N
150 INPUT"[DOWN]TAKEN HOW MANY AT A
    TIME";D
160 IF D>N THEN PRINT"[DOWN]TOO MANY!"
    :GOTO 230
170 P=1:C=1
180 FOR J=N-D+1 TO N:IF 1.7E38/J>=P
    THEN 200
190 PRINT"[DOWN]> 1.7E38
    PERMUTATIONS!!":GOTO 230
200 P=P*J:NEXT:FOR K=2 TO D:C=C*K:NEXT
210 PRINT"[DOWN] NUMBER OF
    PERMUTATIONS =";P
220 PRINT"[DOWN] NUMBER OF
    COMBINATIONS =";P/C:PRINT
230 END
```
END

Factorials: In mathematics, the factorial of N is defined as the product of all the integers between 1 and N. Commodore computers don't have a built-in factorial calculator, but you can easily compute them with the following routine, where N is the number whose factorial you need.

```
100 F=1 : FOR J=1 TO N : F=F*J : NEXT
```

After execution, F will contain the factorial of N.

A. W. Grym
London, England

Roots: BASIC'S SQR function will easily find the square root of any number that the computer can handle. But, what if you need a cube root or other root? Just use this simple expression, where N is the number whose root you want and R is the root you're looking for.

X=N ↑ (1/R)

For example, to derive the fourth root of 32768, enter:

PRINT 32768 ↑ (1/4)

Julia Brennan
Freehold, New Jersey

Logarithms: Most of us who know about logarithms are accustomed to common logarithms, or logs to the base 10. But BASIC's logs are natural logarithms, or logs to the base e (e=2.71828). Natural logs are widely used in electrical engineering and other fields.

If you want to work with common logs, you can define a function that will make the conversion for you, as follows.

```
10 DEF FNL(N)=LOG(N)/LOG(10)
```

Once line 10 has been executed, FNL(q) will give the common log of q, where q is any numeric constant or expression.

Continued on pg. 10

Tips & Tricks

Continued from pg. 8

Calculating FNL(10000) will return a value of 4, which, of course, is the common log of 10,000.

EXP, a BASIC function closely associated with LOG, calculates the value of e raised to the power inside the parentheses following the EXP. If you need a similar function using base 10 rather than e, you can use:

```
20 DEF FNE(N)=EXP(N*LOG(10))
```

You can test it by getting FNE(4), or 10 raised to the fourth power. If you know your high school math, you know that that's 10,000.

Richard Bowman
Basking Ridge, New Jersey

Joystick Port Connectors: It's sometimes useful to connect things other than joysticks to these ports. Radio Shack makes a Joystick Extension Cord (#276-1978) which is very useful at these times. You can use it as is for an extension cord, or you can cut it apart and connect your non-joystick device to it directly.

The ports themselves take a connector widely used in the electronics industry, called the DB-9 Subminiature D. Radio Shack has a variety of these connectors in various configurations, complete with hoods and other cable protectors. Electronics parts stores have even larger varieties. If you buy one of these connectors, be sure to check their fit in your computer. There's never a problem with the connector itself, but sometimes the shell and/or hood will interfere with the computer's cabinet or with the other joystick port.

Mike Swords
Kansas City, Missouri

DIN Connectors: The round connectors used for video and serial bus connections on Commodore computers are called DIN connectors, after the German organization responsible for their design: Deutsches Institut fuer Normung is German for German Institute for Standards. It's commonly referred to by its initials.

If you buy connectors and wire, it's not difficult to make up cables or extension cords for the various DIN connections in your system. I've made serial bus extenders up to ten feet long, without ill effects from the extra length.

I cannibalize the wire for these from Radio Shack's #276-1978 Joystick Extension Cord, which includes ten feet of very flexible nine-conductor cable. Some of the connectors are also available from Radio Shack, but others need to be bought from an audio or electronic supply store.

When making up DIN cords, you must be *very* careful about the numbering of the pins. Pin numbers are usually molded into the connectors themselves, but the digits are very small. The numbering system is often not what you'd expect, and even published diagrams have been known to contain errors. The moral of all this is to be very careful, and to rely on the numbers molded into the connectors.

Another confusing point is that published pinouts sometimes don't say which side of which side of the connector they are illustrating. Commodore manuals show the solder terminal end of the male connector, but others may vary. Again, be very careful.

Steve Macedo
Marietta, Ohio

User Port Connectors: The connectors that mate with your computer's user port are called printed circuit board edge connectors. Edge connectors are made in such a huge variety of grades and configurations that finding the right one can be a real challenge. The one for the Commodore user port is not a very common configuration, which can make your hunt even more difficult.

You can make a serviceable user port connector by taking a hacksaw to a Radio Shack #276-1551. Cut off a 13-pin section from one end of this 22-pin connector, then use pliers to pull out the 13th pin (that pin's former space now becomes the end stop for the cut-down 12-pin connector). Using a similar technique you can cut six pins off the other end, making a nice cassette port connector.

If you'd rather buy an exact-fit connector in the first place, you'll need to find a well-stocked electronics parts store. Ask them for a 12-position dual-sided connector with .156 inch pin spacing, suitable for a 1/16 inch PC board. If they stock this size, they may have it in a variety of types and grades. Just pick the one that fits your taste and pocketbook.

Andy Carlson
Youngstown, Ohio

Connector Polarization: Wherever a PC board edge connector is used, the possibility exists to insert it upside down and do horrible damage to your system. If you look closely at your user port and cassette port, you'll see that Commodore has cut slots in the PC board between certain pins. These are called polarizing slots and prevent upside down insertion of the edge connector. To use them, get polarizing pins for your connectors and insert them between the pins in the appropriate places.

The polarizing pins are tiny pieces of plastic or fiberglas that fit in slots between the electrical pins on the connector. When the polarizing pins match the polarizing slots, the connector can be inserted without difficulty. But if you try to insert it upside down, the pins and the slots won't match, which keeps you from making the connection. If you don't have polarizing pins for your connector, you can often make them yourself from thin plastic.

Lucy S. Terrier
Alton, Illinois

Connector Labels: Like most computers, mine sits on a desk underneath an overhanging shelf, attached to a rat's nest of wires and cables. It's hard to see the back panel at all, let alone well enough to plug in another wire. So when I need to work with the plugs, it's very hard to locate them.

I've solved the problem by labeling each connector on the top surface of my computer in the back immediately above the connector itself. Now when I need to locate the serial bus or some other connector, I know it's right below its label. Once I have it located so precisely, it's easy to replace it by feel alone.

Jim Baggs
Casper, Wyoming

Expansion Port Connectors: Connectors for the Commodore expansion port (the one used for cartridges) are not very easy to find in stores. Also, when you find them, they tend to be expensive. So, whenever I need such a connector, I cannibalize it from a defunct games cartridge. Not only are these widely available, but the cartridge case makes a nice chassis for whatever gadget I want to connect to the port.

If you do buy your own connector, be very careful about pin

Continued on pg. 12

Tips & Tricks

Continued from pg. 10

numbering. Commodore's system of pin numbers is different from that used in industry (pin 1 in the standard system is pin 22 on the expansion port, and so forth).

Biff Sander
Marietta, Ohio

Look Before You Type: Commodore's screen editor makes it very easy to duplicate lines in a program or to create slightly modified lines. All you have to do is list the line in question, change the line number and anything else that's necessary, then press RETURN. Both the new line and the old one are now in memory for you to use.

You can use this to your benefit when typing in programs from this magazine. Many times, entire sections of such programs are virtually identical to one another, and may lend themselves to duplication or semi-duplication by the method described above. Before starting to type a program, you should make a quick search for these sections. Doing this will save you time and will prevent some finger errors. But best of all, it will replace some of typing's tiring drudgery with a more satisfyingly creative method of data entry.
Walt Latocha
Oak Park, Illinois

Temporary Line Deletion: If you want to see the effect of removing a certain line from your program, use the INST key to open up three spaces after the line number, then type REM into the opened space. When you press RETURN, the line will be changed to a remark. If the change doesn't give the effect you want, you can reinstate the line by using the DEL key to delete the REM, then pressing RETURN again. If the change does improve your program, delete the line by typing its number then pressing RETURN.
Charlene Moffitt
Philadelphia, Pennsylvania

Indented Listings: Some versions of BASIC allow you to put spaces between the line number and the rest of the line to indent the line when it's listed. This ability can make your listings easier to understand, such as when you indent all the lines contained in a FOR-NEXT loop.

If you put such spaces into a Commodore BASIC line, the computer will automatically remove them, *unless* you know the trick. The trick is to type the line number, then any shifted letter, then the number of spaces you want to skip, then the statements you want to have in your program line. When you list the line, the shifted letter will have disappeared, but the subsequent spaces will remain. Execution of the program will not be affected in any way.

You can take this one step further to produce a line that is blank except for its number. Type the line number, a shifted letter, a single space, then another shifted letter.
Tom Holleran
Columbus, Ohio

Magazine Placekeeper: When typing in programs from a magazine, use a Reader Response Card to mark the line that you are typing. Even better yet, use one of the subscription cards that fall out whenever you hold the magazine.

Jeff Johnson
Allensville, Pennsylvania

Magazine Entry Hints: Most programs in *Commodore Magazine* include an apostrophe and four letters at the end of every line. You should *not* type those characters unless you're using the Magazine Entry Programs described in the back of every issue. The Magazine Entry Programs are used to insure accurate typing, and they treat the five extra characters as a checksum. The Magazine Entry Programs delete the extra characters from the lines you type, so they don't end up in your program. If you *don't* use the Magazine Entry Program, but do type the extra characters, they are made a part of the program you're typing, where they do nothing but cause Syntax Errors.

When typing in programs myself, I use the Magazine Entry Program-128 for both the 128 and 64 programs. I redefine the F7 key to print an apostrophe, which relieves me from making the awkward SHIFT 7 keystroke. Also, since most of the checksums for DATA statements start with the letter B, I redefine the F5 key as apostrophe-B, which further eliminates keystrokes.
John B. Boniger
Rock Island, Illinois

Magazine Entry Improvements: I've modified my Magazine Entry Programs to eliminate the need for remembering the NEW and SYS commands. For the C64 version, just remove the END statement from line 120, then add these new lines.

```
130 PRINT "[CLEAR,DOWN4] NEW"
140 PRINT "[DOWN2] SYS49152"
150 POKE 631,13 : POKE 632,13 : POKE 634,13 : POKE
198,3 : END
```

Continued on pg. 14

THE LEADING C64 WORD PROCESSOR JUST GOT BETTER!



No Brag. Just Fact. WORD WRITER 3 has more features, more power, and is easier-to-use than any other C64 word processor.

And now the facts! **We've added:**

- **An 85,000-Word Spell Checker** – plus, unlimited sub-dictionaries.
- **An Integrated Thesaurus** with over 60,000 synonyms and alternatives.
- **An Integrated Outline Processor** that quickly organizes notes, facts, and ideas into a convenient outline format.
- **An 80-Column Print Preview Mode**
- **Highlighting:** Prints out your text incorporating underlining, **boldface**, *italics*, superscript, subscript, and more.
- **Headers and Footers**
- **Automatic Program Set-Up:** Configures WORD WRITER 3 to your choice of printer codes, screen colors, and more.
- **SwiftKeys*** access commands quickly, using a minimum of keystrokes.
- *. . . and much, much more!*

You Get Free, Ongoing Technical Support

Rest assured. When you show your support by buying Timeworks software, we never stop showing ours. That's why our twelve Customer Support Technicians have been giving our T.L.C. (Technical Loving Care) *for over four years.* And, all our programs have a Money Back Guarantee**

Word Writer 3 interfaces with Timeworks' other C64 programs for a complete Productivity System:



- **DATA MANAGER 2 with Report Writer**
  A highly flexible filing and recordkeeping system that stores, retrieves, sorts, evaluates, and updates large amounts of information. Includes: Report Writing, Graphics, Statistics, and Label Making capabilities.
- **SWIFTCALC with Sideways**
  A powerful, surprisingly easy-to-use electronic spreadsheet for home and business use. Plus, with Sideways, prints all your columns on one, continuous sheet – sideways!
- **PARTNER 64**
  A cartridge-based product with eight instantly accessible desktop accessories. PARTNER operates *"concurrently"* with your other C64 software programs.

Available for Commodore 64/128* Computers (64K, 40 Column)

Suggested Retail List Prices:

WORD WRITER 3	**$49.95**
DATA MANAGER 2	**$39.95**
SWIFTCALC	**$39.95**
PARTNER 64	**$59.95**

Now at your favorite dealer, or contact Timeworks today.

To Order Call
1-800-535-9497

These Programs Are GEOS* Compatible

TIMEWORKS
More power for your dollar

Timeworks, Inc. 444 Lake Cook Road
Deerfield, Illinois 60015 312-948-9200

** Details on every Timeworks package.
* Registered trademarks of Commodore Electronics, Ltd., Timeworks, Inc., Berkeley Softworks, Inc.

Tips & Tricks

Continued from pg. 12

For the C128 Magazine Entry Program, remove the END statement from line 120, then add the following lines.
130 PRINT "[CLEAR,DOWN4] NEW"
140 PRINT "[DOWN2] SYS4864"
150 POKE 842,13 : POKE 843,13 : POKE 844,13 : POKE 208,3 : END

The new lines use the dynamic keyboard technique to execute the NEW and the proper SYS call.

Dale E. Roenneburg
Brodhead, Wisconsin

Hang-up Recovery: Recently I had a program that wouldn't run on my C64. When I tried it on a friend's computer and it worked perfectly, I thought I'd have to send my own machine to the repair shop. On investigating further, I found that my voice synthesizer, connected but not enabled, was causing the program to hang up. As soon as I unplugged it, the program worked okay.

When I told the story to my computer friends, many of them had had similar situations involving printers, interfaces and the like. So my advice is this—when you can't get a program to work or when certain peripherals seem to hang up, try removing all unnecessary peripherals from your system.

Steve Ward
Tacoma, Washington

Hidden GEOS Characters: Not mentioned in the GEOS manual is the existence of several characters not normally available from the C64 keyboard. These characters are part of the standard ASCII character set, but not part of the Commodore set. You'll get them by pressing the indicated keys with the Commodore key held down.

Left curly bracket	Commodore and colon
Right curly bracket	Commodore and semicolon
Underline	Commodore and dash
Backslash	Commodore and slash
Vertical line	Commodore and up arrow
Tilde	Commodore and asterisk
Left apostrophe	Commodore and commercial at

Francis Volpe
Pittsburgh, Pennsylvania

Super Huey Hint: I love this program, but it's been frustrating to wait so long for the engine to warm up. One day I discovered that if I brought the RPM's down to 120, the initial warm-up time was reduced by over one-half.

Michael Schroeder
Hudson, New York

C64 Mirror: This program will definitely get the attention of the next person to use your C64. Just before you turn your computer over to the next person in line, load and run this program.

Frank Colaricci
Winter Park, Florida

C64 Mirror

```
10 REM C64 MIRROR - FRANK COLARICCI
15 PRINT CHR$(142);CHR$(8)
20 A=828:REM RELOCATABLE
25 FOR I=A TO A+87:READ B:POKE I,B
30 C=C+B:NEXT
35 IF C<>11844 THEN PRINT"ERROR":END
40 SYS A
45 DATA 173,014,220,041,254,141,014,
   220
50 DATA 165,001,041,251,133,001,169,
   000
55 DATA 133,251,133,253,169,208,133,
   252
60 DATA 169,048,133,254,162,008,134,
   250
65 DATA 160,000,177,251,162,008,106,
   038
70 DATA 002,202,208,250,165,002,145,
   253
75 DATA 200,208,239,230,252,230,254,
   198
80 DATA 250,208,231,165,001,009,004,
   133
85 DATA 001,173,014,220,009,001,141,
   014
90 DATA 220,173,024,208,041,240,009,
   012
95 DATA 141,024,208,169,048,133,056,
   096
```

END

C128 High-Resolution Helper: Using the C128's graphic modes is great, but it can be a pain to figure out the numbers corresponding to various screen locations. This little program makes that work a lot easier, by letting you move a tiny pixel cursor around the screen while it reports its own location.

Just load and run the program, then use the numeric keypad to move the little cursor. The 8-key moves it up, the 2-key moves it down, and so on. To leave a dot in the cursor's current position, press the 5-key. Press STOP to exit the program.

If you delete line 5010, you can use the program in conjunction with other graphics programs you are developing. Just insert Hi-Res Helper, less line 5010, after the other program's DRAW statements. You'll have the little cursor plus whatever your program has drawn.

Jerry W. Jarvis
Spokane, Washington

Hi-Res Helper

```
5000 REM HI-RES HELPER - JERRY JARVIS
5010 GRAPHIC 1,1:TRAP 5180
5020 X=160:Y=100:GOTO 5160
5030 DRAW 1,X,Y
5040 GET KEY J$:J=VAL(J$)
5050 IF J=5 THEN F=1
5060 IF F=0 THEN DRAW 0,X,Y
5070 IF J=1 THEN Y=Y+1:X=X-1
5080 IF J=2 THEN Y=Y+1
5090 IF J=3 THEN Y=Y+1:X=X+1
5100 IF J=4 THEN X=X-1
5110 IF J=6 THEN X=X+1
5120 IF J=7 THEN Y=Y-1:X=X-1
5130 IF J=8 THEN Y=Y-1
5140 IF J=9 THEN Y=Y-1:X=X+1
5150 IF J<>5 THEN F=0
5160 CHAR 1,1,1,"X="+STR$(X)+"
```

Continued on pg. 58

Is Getting The Answer To Software Problems A Bigger Problem Than The Problem?

Don't stay on hold when there's help online from CompuServe® Software Forums.

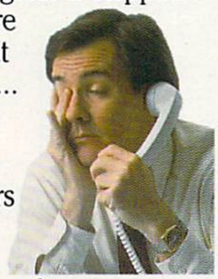


The new upgraded version of your software locks up. And every time you reboot, you get stuck in the same place in the program.

You've chucked the manual, because you've done exactly what it tells you to do six times already. So you call the software company.

Now you spend half a day beating your head against a brick wall of busy signals, ranting at recorded messages, hanging around on hold. And you still don't get the solution to your problem.

Meanwhile, progress is stopped and your profits are dribbling away. But wait. There's help...



Several prominent, progressive software publishers recognize this problem, and working with CompuServe, have developed a solution—CompuServe Software Forums.

Now you can go online with experts from the companies that produced your software and get prompt, written answers to your specific problems. You can even talk with the actual software developers.



Aldus®, Ashton-Tate®, Autodesk®, Borland International®, Creative Solutions®, Digital Research®, Living Videotext®, Lotus® Inc., Microsoft®, MicroPro®, Misosys Inc.® and Software Publishing® all have CompuServe Software Forums. And we keep adding more.



CompuServe's large subscriber base also puts you in touch with thousands of other, often more experienced, users of the same software. You'll find they can give you lots of creative ways to get the most out of your software.

And software forums are the best way to learn about product updates, new product announcements, new ways to expand the uses of your software, and offer free uploads of your own programs.

Our online electronic magazines frequently publish software reviews. And you can find help for many other software products in our other computer-related forums for IBM®, Tandy®, Atari®, Apple®, Commodore®, TI® and others.



The last thing you need when you've got a software problem is a bigger problem getting answers. So, from now on, get prompt, informed answers on CompuServe Software Forums.

To buy your CompuServe Subscription Kit, see your nearest computer dealer. Suggested retail price is $39.95.

To order direct or for more information, call 800-848-8199 (in Ohio, 614-457-0802).

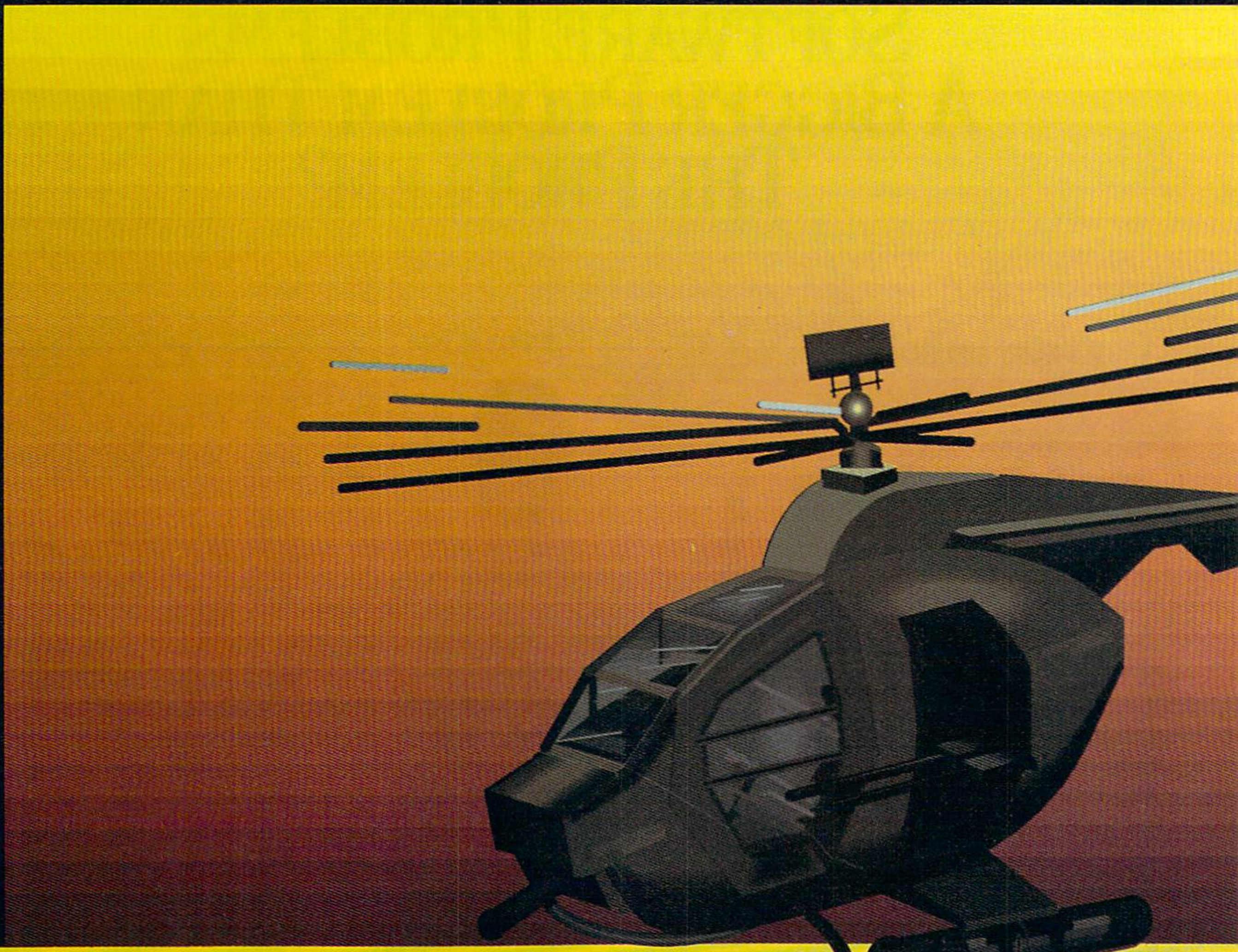
If you're already a CompuServe subscriber, just type GO SOFTWARE at any ! prompt.



CompuServe®

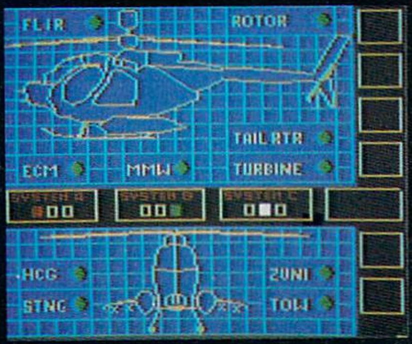
Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, OH 43220

An H&R Block Company

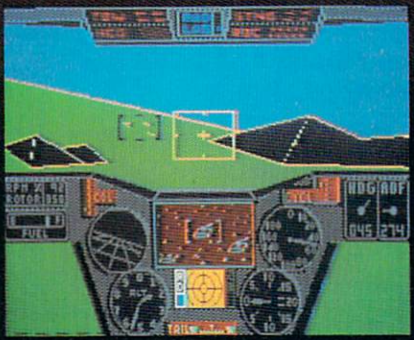


In a market full of helicopter simulations like Super Huey II, Gunship, and Infiltrator, it's nice to find a product like ThunderChopper that flies high above the rest!

Colonel Jack Rosenow USAF (Ret.), President of ActionSoft Corp., has the experience to provide all of the helicopter action and strategy you've been looking for! ThunderChopper incorporates the most advanced graphics, flight systems, and game-playing factors to provide a sensational balance of strategy and fun:



- Action-packed animated graphics include real 3D scenery and airborne threats. The competition's graphics just don't compare.
- A sophisticated instrument panel lets you scan all vital information at a glance whether performing combat, exploration or rescue operations. ThunderChopper's advanced instrumentation includes Forward-Looking Infra red, CO2 laser radar, zoom television, and ECM. Armament includes TOW and Stinger missiles, a Hughes Chain Gun, and Zuni rockets.
- Better program and documentation design gets you up flying exciting combat missions in minutes.



As Colonel Jack says: "ThunderChopper is the ultimate in helicopter action and realism. Nothing else even comes close. No other simulation can boast this much fun!"



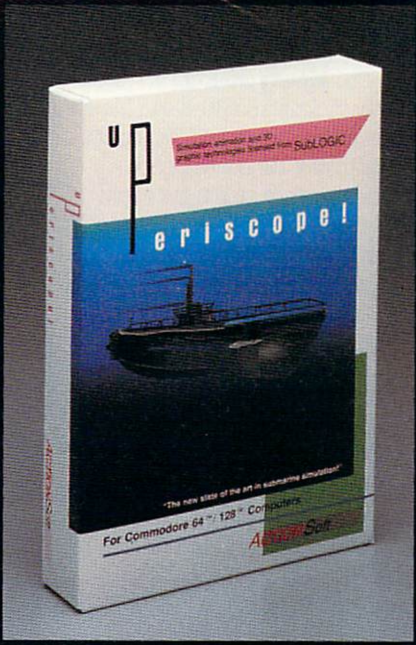
ThunderChopper

Simulation animation and 3D graphic technologies licensed from **SubLOGIC**

Up Periscope!

The new state of the art in submarine simulation. The superior strategic play action and 3D animated graphics of this simulation put it generations ahead of the pack.

See Your Dealer...
Or write or call us for more information. ThunderChopper and Up Periscope! are available on disk for the Commodore 64/128, Apple II, and IBM PC line of personal computers for the suggested retail price of $29.95. For direct orders please specify which computer version you want. Include $2.00 for shipping and specify UPS or first class mail delivery. Visa, MasterCard, American Express, and Diners Club cards accepted.



- $29.95 -
Better Engineering at a Better Price

ACTIONSoft
"GENERATIONS AHEAD IN STRATEGY ACTION SOFTWARE"

122-4 S.RACE ST. URBANA . IL 61801
(217) 367-1024

S I L I C O N   V A L L E Y   I N S I D E R

BY MATTHEW LEEDS

From Hardhat to the Video Title Shop

Previews of new products from the Valley.



WILSON HARP

Brown-Wagh Publishing has released an upgraded version of **Analyze**, their spreadsheet for the Amiga. The new version now has macro support, letting you build your own templates and keystroke shorthands. It also offers sorting of all data types, file icons for Workbench support, and graphics.

The new graphics functions include bar, pie and area graphs in 2D and 3D. All graphs can be saved in IFF format and can be enhanced with text. A mode that changes the graphs as you change the data is also available. You can have up to four graphs on the screen at one time.

As I mentioned in an earlier column, Aegis Development has arranged to publish **Musicraft** under the name of **Sonics**. This upgraded version has features like full MIDI support, control of up to eight internal voices, Kickstart 1.2 compatibility, and an expanded library of instruments and scores. The entire program runs smoother and cleaner, and uses standard Aegis file requesters.

Last September I met Vladimir Schneider, President of **Zirkonics Corporation,** a software development house in Montreal. He was demonstrating a new text editor for the Amiga called **PTE**. Text editors are not word processors—they enter and edit text, but don't output to a printer. If you write only short letters, a word processor is useful, but if you write long documents or program code, a text editor is the tool of choice.

PTE lets you create your own editing environment. You decide what macros you want and what key combinations will call them up. You can have as many files in memory as you have space for, and can copy and move text between them. A powerful program.

Sedona Software has produced a personal finance manager package for the Amiga called **Money Mentor**. Written completely in AmigaBASIC, it can handle up to 30 separate accounts and 200 budget categories in any 12-month period. Transactions within accounts or split payments are supported, as is check printing and several search options. There are also quite a few report options including color graphs or printed reports.

I have mixed feelings about this program. It has most, if not all, of the functions you need in a personal finance package, and it is easy to use. However, it does not follow the Amiga standard in several areas of operation, and is very slow updating the screen. It would benefit from being run through a BASIC compiler once one is available for AmigaBASIC.

Electronic Arts has picked up distribution of the Cygnus line of software. One of the new releases is **Quizam** for the 64. This is the most comprehensive computer trivia game I've seen. It not only allows up to eight players to participate, but offers you the option to create your own questions and answers. The entire program is based on the premise that you and your fellow players are part of a repair crew working on satellites that are failing. Instead of requesting the normal entry codes, they have begun to require odd, random, trivial information. It's a cute premise.

Also from Electronic Arts is the 64 version of **Moebius**. I don't get much time to play adventure games, but I was very taken by the fluid animation of the combat scenes in this oriental fantasy role-playing game. I was also intrigued by the black headband included in the gamebox. Once I put it on, I began to feel different, more cat- like in my movements. It may take me a while to become a true dicipline of Moebius the Windwalker, but I'm working on it.

Illusion Software has just released their first set of fonts for the Amiga, **Illusion Fonts #1**. This is the beginning of a library of fonts, with ten complete fonts in a variety of sizes from 8 to 21 points. The fonts can be used in any of the

Continued on pg. 98

FLIGHT!

From the sophisticated realism, detail, and intellectual stimulation of Flight Simulator...



...to the brute-force fun, thrills and excitement of Jet...

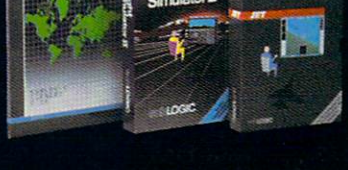


...with new adventures in Scenery Disks...



...SubLOGIC. The State of the Art in Flight.

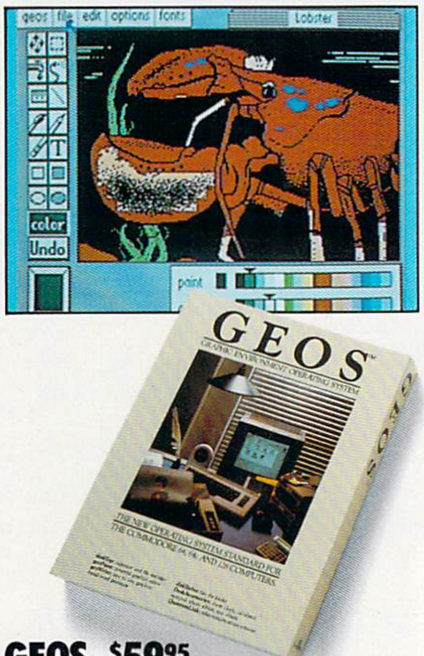
See Your Dealer. For additional product ordering information or the name of the dealer nearest you, call (800) 637-4983.



subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

ORDER LINE: (800) 637-4983
(except in Illinois, Alaska and Hawaii)
Open 7 AM to 9 PM Central Time

IF YOU STILL THINK CO
PLAY AROUND

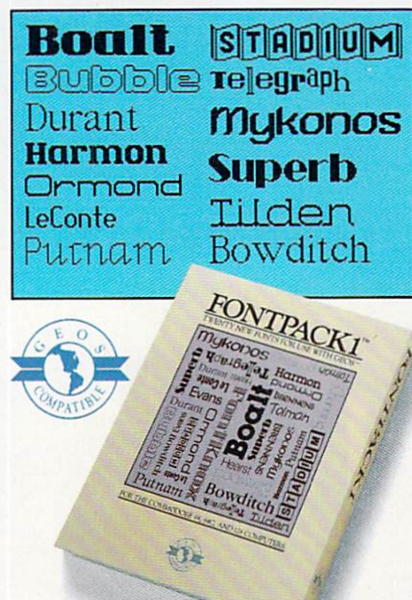


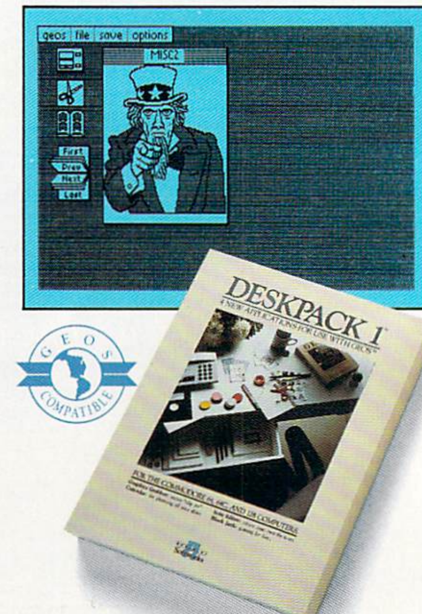
GEOS $59.95

The Graphic Environment Operating System that opens up a whole universe of new possibilities for Commodores. With geoWrite, geoPaint, fast-loading diskTurbo and support for all GEOS-compatible applications.

FONTPACK 1 $29.99

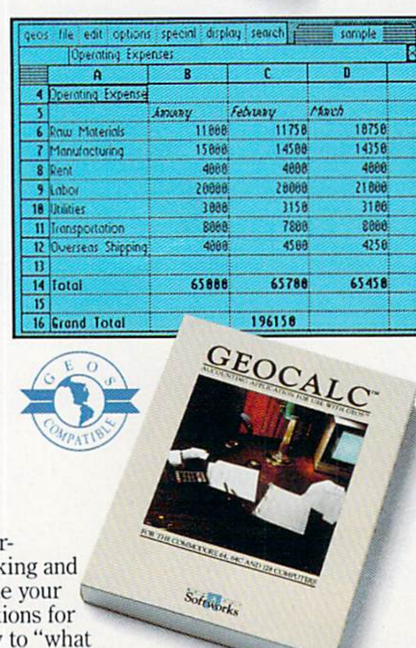
A collection of 20 more fonts for use with GEOS applications, in various shapes and sizes for more expressive and creative documents.

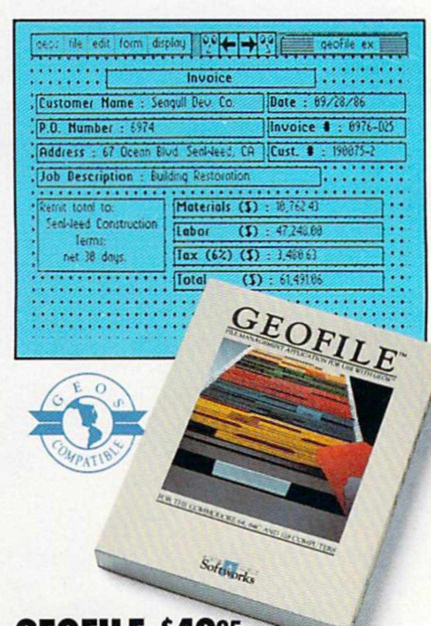




DESKPACK 1 $34.95

Four GEOS-compatible applications: Graphics Grabber for importing art from Print Shop,™ Newsroom™ and Print Master™ graphics; Calendar; Icon Editor and Black Jack Dealer.





GEOCALC $49.95

The GEOS-compatible, number-crunching spreadsheet for tracking and analyzing numerical data. Create your own formulas, perform calculations for anything from simple geometry to "what if" cost projections.

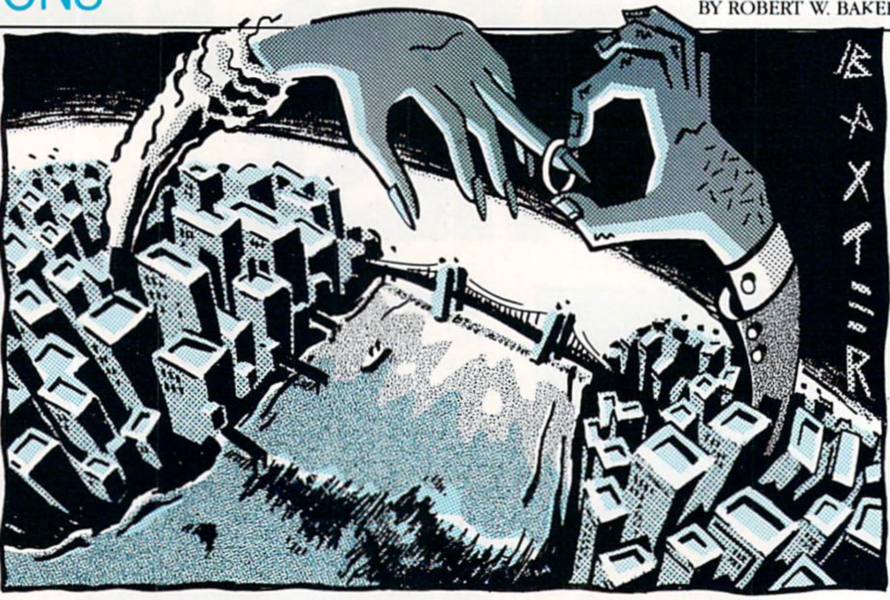
GEOFILE $49.95

The GEOS-compatible database manager that sorts, edits and prioritizes whatever data you feed it. You fill out the "input form," specify your command, and geoFile takes it from there.

BY ROBERT W. BAKER

Inside QuantumLink

Explore the inner workings of the QuantumLink telecommunication service with network pro Bob Baker.



In my first column, I mentioned some of the utilities commonly used to make the uploading and downloading of files somewhat easier. The more popular of these utilities are ARC, LYNX and LIBRARY, which are used to compress or combine files. Since that time, it has come to my attention that several of the early versions of the LYNX utility had a built-in protection scheme that could cause problems if the program was modified.

Supposedly, the LYNX utility did a checksum to verify that certain portions of the program were not modified. If the computed checksum did not match the expected value, the utility could wind up destroying files on your disk. The later versions of the LYNX utility (version 9 or later) have had this quirk removed, and all earlier versions of LYNX were to be deleted from the QuantumLink download libraries.

If you haven't done so already, I would strongly recommend downloading a copy of the latest version of the LYNX utility. While you're at it, you might also want to check if you have the latest version of ARC and other utilities that you might be using. There are new versions being uploaded quite frequently, so you might want to check them from time to time.

If you find the utilities helpful, remember that most are freeware programs and the authors depend on donations to fund work on improvements and future development. Also, remember that some authors of freeware utilities might provide added documentation, companion programs, or update modifications to registered owners. So you might find it worthwhile to support a particular utility you find especially useful.

Hot Off the Wire

The CitiLink section of News & Information is now off to a running start. However, QuantumLink is looking for more input from everyone. The various "city editors" have been doing a fine job, but more input is needed from other users in each area. If you have something to share, be sure to drop by and let others know what's going on.

MatchMaking seems to be getting pretty successful in finding friends as well as lovers. There are even Qweddings these days, which are mock weddings that are held on-line on QuantumLink. However, some Qweddings are very serious and the Qcouple want to be known on-line as Qhusband and Qwife. Others are simply friends who just enjoy sharing time together on-line.

Steever and Loriannel went even further. They met on-line on Q-link in Ed's Pub back on Labor Day last year. If you haven't dropped by, Ed's Pub is a popular public room in Q-link's People Connection. Anyway, since they met, Steever and Loriannel have spent a lot of time together on-line. Even though Steever lives in Texas and Loriannel lives in Alabama, they fell in love.

After talking a lot on Q-link, Steever and Loriannel decided to go a little further and finally talked over the telephone. From there they have come a long way, meeting in person, and finally getting engaged.

On December 31st they were Qmarried in the Auditorium on Q-link, with Satin S8 (a veteran of over 40 weddings on Q-link in the past year) performing the ceremonies. Then, later that same night, Steever and Loriannel were to get married in a real ceremony to be held in Birmingham, Alabama. Good luck Steever and Loriannel, and I hope you continue to enjoy Q-link and the other people you've met on-line.

If you're interested in checking out the MatchMaker section or finding out more about what's going on in People Connection, you'll find that and more in the P.C. Society section of the Just For Fun area of Q-link. There you'll find The Ear, Room Notices and Special Events sections, where you can find out the latest gossip and what's going on in People Connection. You'll also find more information on the Qshorthand graphics and abbreviations I mentioned in my February column.

While you're there, be sure to also check out the Photogallery. This is where you can download graphic pictures of other users and find out how to submit a photo of yourself, so your on-line friends can see what you look like. All the information on how to submit photos and how to view a photo after you've downloaded the data file is all provided in this area as well.

Actually, the Just For Fun section of Q-link is probably one of the largest sections of the system these days. Besides P.C. Society, you'll also find the Clubs & Special Interest, Sports Center, Fun & Games, Treasure and Rocklink sections.

In the Clubs & Special Interests section you'll find the renovated Crossroads Cafe, where you can stop by for a lively debate or interesting conversation. The Club House, Game Room, Support Center, and Work Forces sections each contain various areas within them that cover a wide range of interests. There you can share information with others about your particular hobby, occupation or special

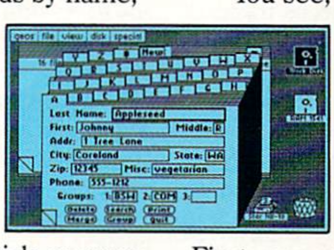
Continued on pg. 98

ARE YOU SEARCHING FOR FRIENDS?



If you're finding it increasingly difficult to find anything at all, maybe it's time you found out about geoDex. The GEOS-compatible directory that generates mailing lists. Prints address labels. And sorts out all sorts of things for your Commodore.

Try directory assistance. With a little help from geoDex, you can call up a directory organized from any three categories you choose. Which means you can list your friends by name, telephone number or almost anything else that can be assigned its own three-character code.



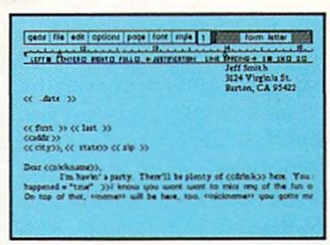
Like "MEN" for guys you know. Or "GRL" for girls you know. Or "FOX" for girls or guys you'd like to know.

But no matter how you choose to categorize them, if you can point and click a mouse, you can call up any list of friends with geoDex's easy-to-read graphics.

Our most inviting feature. Of course, once you've gotten your friends organized, the next thing we recommend you do with geoDex is really very simple.

Throw a party.

You see, geoDex comes with geoMerge, a mail merge program that customizes form letters, announcements—even party invitations—with the names and addresses stored in geoDex. First you write the letter with geoWrite. Then you select a list from geoDex.



Put them both together with geoMerge and it's toga time!

The search is over. So if you're tired of looking for friends, waltz right down to your software dealer and ask him for geoDex. We can't guarantee it'll win you more friends, but it'll certainly keep you from losing them.

To order call 1-800-443-0100 ext. 234

geoDex $39.95

(California residents add 6.5% sales tax.)

$2.50 US/$5.50 Foreign for shipping and handling. Allow six weeks for delivery.

Commodore is a trademark of Commodore Electronics, Ltd. GEOS, geoDex and Berkeley Softworks are tradenames of Berkeley Softworks.

GEODEX GEOS COMPATIBLE



Berkeley Softworks

The brightest minds are working at Berkeley.

BY SHAY ADDAMS

AutoDuel vs. Roadwar 2000

News and opinion from a leading explorer of those fantasy realms called adventure games.

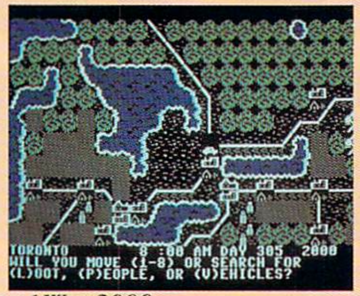
Besides sharing a common premise—a futuristic America where people wage war in armed cars—this pair of role-playing adventures has something else in common: They appeal to aficianados of other forms of computer gaming as well as to hard-core adventure fans. *AutoDuel*'s quest is built around an arcade-style action game, while *Roadwar 2000*'s tactical elements will attract the strategy/war gamer. I like them both, but I suspect most people will choose one over the other. So let's kick the tires and check under the hoods to find out which game is best for you.

AutoDuel

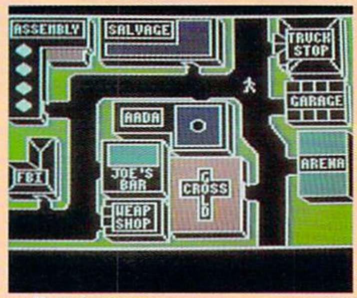
Programmed by Lord British and Chuckles, *AutoDuel* is based on a popular board game called Car Wars™. It takes place on the United States' Northeastern corridor in the year 2030. After a series of disasters, the nation is in chaos: Outlaws roam the highways and the autoduel—which is like a demolition derby with guns—has replaced baseball as the national pastime. You direct the actions of a character who is created by distributing 50 points among three skills: driving, marksmanship and salvage.

The action begins on foot in New York, where an animated figure moves around an aerial view of the city's buildings and streets. Your first goal is to earn enough money to buy a more powerful car. This is accomplished by winning an Amateur Night competition in a car provided by the arena, or by hopping a bus to Atlantic City and getting lucky at poker or blackjack. Your character's skills improve a bit with each victory, in addition to gaining prestige points and money.

From an aerial view, the duels are depicted with scrolling animation but no sound effects, and gauges show things like speed and radar display. A clever trick incorporates the role-playing aspects of the game into the arcade sequences: the higher your driving skill, the more responsive the joystick. As your marksmanship improves, you stand a better chance of hitting the enemy vehicles. The battles offer new thrills for shoot-'em-up enthusiasts: Mount weapons in the appropriate locations and you can fire from the front, rear or either side of the car. These effects are also well-animated, particularly the flamethrower.



Road War 2000



AutoDuel

When you've saved enough cash, you can buy a customized car, van or truck. In an assembly plant, you choose a chassis whose weight and space determine things like the size of the engine you can use, the kind of weapons, and the amount of armor. The design process is set up like a joystick-operated spreadsheet that automatically keeps tracks of all the variables, including money, so you don't have to do any math. The variety of weapons—oil slicks, lasers, recoilless rifles—and various tires, engines and other parts enable you to design the death car of your choice. And you can own up to 20 vehicles, storing them in garages in the game's 16 cities.

If you get tired of fighting duels, you can earn a living as a vigilante who tracks down outlaws on the road or as a courier for the American AutoDuel Association. The higher your prestige rating, the more valuable the courier missions you'll get from the AADA. By travelling to distant cities, you'll discover clues to a series of mini-missions. These are uncovered when you enter the bars and truck stops and pick up rumors. Inside a building, menus appear to offer various options: accept a courier mission, buy a drink, and so on.

Eventually you are assigned the ultimate goal: to track down a major crime boss. However, this is the game's weak point—you don't have to solve the mini-quests along the way in order to qualify for the main goal. (Imagine being able to enter *Ultima IV*'s Abyss without having first found the bell, book and candle.) So the puzzle-solving aspects of the game have been deemphasized in favor of action and a fair amount of role-playing.

Still, *AutoDuel* kept me at the wheel for weeks—and the fun's not over when you complete the final mission, for you can still hop in your car and hit the road or arena for a battle. It's available for the Commodore 64 and Amiga.

Roadwar 2000

Roadwar 2000 presents a similar scenario, except that you must find eight good guys instead of one bad guy—these good guys are the scientists whose work will eliminate an epidemic sweeping a land also devastated by nuclear bombs, foreign invaders, and a host of domestic troublemakers. A single car represents your gang as you press keys to move it across a map of southern Canada, the United States and northern Mexico. Only a portion of the map is seen at any one time, and the whole thing covers about ten screens.

Your gang starts small, but may grow to include hundreds of members. In addition to developing their skills, you command up to 15 vehicles whose attributes—speed, maneuverability, armor and so on—must be enhanced by finding body shops and garages. There are 19 kinds of cars, trucks and buses, but you don't get to buy or design vehicles as in *AutoDuel*. And they are outfitted with crossbows or generic firearms—nothing exotic.

In the 120 cities or while traveling the highways, your options are usually limited to three: hit L to loot for food, ammo and other stuff; V to search for vehicles; and P to look for people. The results are randomly determined and displayed in brief messages shown in a three-line text window below the map.

When looking for people, you may find a scientist, a clue, or you might run

Continued on pg. 89

BY MARK HAUGAN

Calendar Maker

for the Commodore PET, VIC 20, 64 and 128



BOB CLARK

Print out a full-page calendar for any month in any year, past of future.

How do you keep your life and activites organized? If you are like me, you need all the help you can get! One practice that helps me to keep my schedule in order is to make note of upcoming events and activities on my calendar. This works well until I have to erase a zillion notes to realign my schedule after a change of plans, or until my calendar gets so full that I simply run out of space for notes. At times like that I wish I could simply zap up another calendar for the month and start fresh or make my additional entries.

Well, now I can do exactly that with Calendar Maker. I hope some of you will find it as useful as I have. With it you can print out a full-page calendar for any month in any year, past or future. That seems a rather powerful thing to be able to do. It is remarkable that such a short BASIC program manages to do it.

The routine that actually computes the day of the week that begins any month in any year is contained in lines 70 and 79. This is the implementation in BASIC of a well-known algorithm that is at the heart of perpetual calendar programs that run on many mainframe systems. To this routine, I have added coding that makes it easy for you to select the month and year you are interested in and then print out a nicely formatted, full-page calendar of that month.

Calendar Maker works for months in the past as well as the future, so you can print out the months in which historical events took place. Did you know that the day on which the Second Continental Congress adopted the Declaration of Independence, July 4, 1776, was a Thursday? Who knows, that question might come up during your favorite trivia game!

If you want to use Calendar Maker in this way, you really need to know a little more about the way in which the calendar has evolved. Calendar Maker prints out the months according to the modern Gregorian calendar. This is the calendar that we all use everyday and that is in use in most countries throughout the world. It is named for Pope Gregory XIII who introduced it in 1582 as an improvement on the Julian calendar which had been in use since 46 B.C., the time of Julius Caesar. Calendar Maker can, therefore, be used reliably for dates in European history after 1582.

For dealing with English and American history, you need to know that the modern Gregorian calendar was not used in England or the American colonies until 1752. I suspect the delay between 1582 and 1752 had something to do with the conflict that existed then between the Catholic and Anglican churches.

Calendar Maker is very easy to use. It should run on any Commodore 8-bit computer: PET, VIC 20, 64 and 128, using just about any Commodore printer or any non-Commodore printer with a suitable interface. The program does print a couple of Commodore graphics characters and does use the Commodore 1525 printer control codes, CHR$(14) and CHR$(15), to turn double width printing on and off in lines 134 and 136. These are the only things that could possibly be misinterpreted by some printers. If you should have any problems, check your printer and interface manuals to see if graphics characters or these control codes should be handled in some special way. As it is, Calendar Maker runs beautifully on my Gemini-10X with Connection interface. This combination emulates a Commodore 1525 printer.

To print a calendar, simply load and run Calendar Maker. You will be prompted for the month and year that you wish to print. You may enter the number for the month, its name, or an abbreviation which consists of enough of the first few letters of the month's name to identify it uniquely. For example, to enter September you would type 9, September, Sept or even S, since no other month has a name that begins with S. Remember the comma between the month and year if you want to enter them at the same time. I have set up Calendar Maker so that if you enter a number less than 100 for the year, say, 85, then Calendar Maker assumes you mean 1985.

Once you hit the RETURN key to enter you choice, Calendar Maker will ask you to make sure your printer is turned on. This gives you a chance to make certain that it is and that you are set at the start of a fresh page. When you are ready, hit any key and Calendar Maker will proceed to fill the page with the calendar for the month you selected. If you should want to print additional months, simply run the program again. Make sure that you set your printer to the start of the next page when you are prompted to check that your printer is on and ready.

One final thing, what day of the week were you born on? The first month you might want to try Calendar Maker on is the one you were born in. As it turns out, I was born on a Sunday. ■

Before typing this program, read "How to Enter Programs" and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Calendar Maker

```
10 REM PRINTING PERPETUAL
   CALENDAR'BAEE
20 DIM C$(42),D$(31),M$(12),N(12),
   DN$(7)'BINE
22 PRINT"[CLEAR]":PRINT SPC(9);"[RVS]
```

Continued on pg. 28

NUMBER ONE ARCADE HITS...



FOR YOUR COMPUTER.

Put on your black belt and challenge your friends or the computer through nine picturesque settings in this leading Martial Arts Game to become the **KARATE CHAMP**. For the Commodore 64™/128 and the 48K Apple II® Series.

As the crack shot **COMMANDO**,* battle overwhelming odds to defeat advancing rebel forces. Armed with only a machine gun and hand grenades, you must break through the enemy lines to reach the fortress. For the Commodore 64™/128.

Prepare for the fight of your life . . . you are the **KUNG-FU MASTER**.** Battle the evil forces through the five dangerous floors in the wizard's castle to rescue the captive fair maiden. For the Commodore 64™/128 and the 48K Apple II® Series.

DATA EAST

DATA EAST USA, INC.
470 Needles Drive, San Jose, California 95112
(408) 286-7074

Continued from pg. 26

```
   PERPETUAL CALENDAR "'DECI
30 REM INITIALIZE DATA MATRICES'BWRF
32 FOR I=1 TO 31:READ D$(I)
   :NEXT I'FMYG
34 FOR I=1 TO 12:READ M$(I)
   :NEXT I'FMHI
36 FOR I=1 TO 12:READ N(I):NEXT I'FLWK
38 FOR I=1 TO 7:READ DN$(I)
   :NEXT I'FMHM
40 REM INPUT MONTH AND YEAR'BRSF
42 PRINT"[DOWN] FOR WHAT MONTH AND
   YEAR DO YOU WISH TO"'BAAM
44 PRINT"[SPACE10]PRINT THE
   CALENDAR?"'BARL
46 INPUT"[DOWN] MONTH, YEAR ";M$,Y
   :IF Y<100 THEN Y=Y+1900'GQTQ
50 M=VAL(M$):IF M>0 AND M<13 THEN
   62'HNEI
52 I=1'BCSE
54 IF M$=LEFT$(M$(I),LEN(M$)) THEN M=I
   :GOTO 62'HTGN
56 I=I+1:IF I>12 THEN 42'FJCM
58 GOTO 54'BCQJ
60 REM PRINTER CHECK'BMRG
62 PRINT"[DOWN2,SPACE2]
   BE SURE YOUR PRINTER IS TURNED ON!"
   :FOR I=1 TO 500:NEXT I'FIUS
64 PRINT"[DOWN,SPACE7]HIT ANY KEY TO
   CONTINUE."'BAKO
66 GET A$:IF A$="" THEN 66'EHTL
70 REM CALENDAR ROUTINE'BPNH
71 J=367*Y-INT(7*(Y+INT((M+9)/12))/4)
   +INT(275*M/9)+1721031'PJEV
72 K=0:IF M<=2 THEN K=-1'HHKL
73 J=J-INT(3*(INT((Y+K)/100)+1)/4)
   'JSWP
74 K=N(M):IF M<>2 THEN 79'FKRM
75 W=INT(Y-100*INT(Y/100))
   :X=INT(Y-4*INT(Y/4))
   :Z=INT(Y-400*INT(Y/400))'SMND
76 IF X<>0 THEN 79'EEQM
77 IF W=0 AND Z<>0 THEN 79'GGTP
78 K=29'BDDM
79 X=J-7*INT(J/7)'FHWR
100 REM'BARU
102 REM PRINTING ROUTINE'BPGC
104 REM'BARY
110 REM PREPARE OUTPUT'BNRA
112 H$="[CMDR @11]"'BCCG
114 FOR I=0 TO 42:C$(I)="[CMDR G,
    SPACE10]":NEXT I'FMQH
116 FOR I=1 TO K:C$(I+X)=D$(I)+"
    [SPACE8]":NEXT I'HRJL
120 REM OPEN PRINTER CHANNEL'BSVC
122 OPEN 4,4'BDAA
130 REM PRINT TITLE'BKHB
132 IF I+X<=36 THEN F5=1:FOR I=1 TO 5
    :PRINT#4:NEXT I'LPPK
134 T$=M$(M)+STR$(Y):C=20-LEN(T$)/2
    :PRINT#4,CHR$(14);'JCAN
136 FOR I=1 TO C:PRINT#4," ";
    :NEXT I'FJMJ
138 PRINT#4,T$:PRINT#4,CHR$(15)
    :PRINT#4'ENML
140 REM PRINT DAY NAMES'BNJD
142 FOR I=1 TO 7:PRINT#4,DN$(I);
    :NEXT I:PRINT#4'GREI
150 REM PRINT MAIN CALENDAR'BRWF
151 FOR I=1 TO 5:J=7*I'FHGF
153 FOR L=1 TO 7:PRINT#4,H$;:NEXT L
    :PRINT#4'GNNJ
155 FOR L=6 TO 0 STEP -1
    :PRINT#4,C$(J-L);:NEXT L
    :PRINT#4,"[CMDR G]"'JTIP
157 FOR L=1 TO 7:FOR N=1 TO 7
    :PRINT#4,C$(0);:NEXT N
    :PRINT#4,"[CMDR G]":NEXT L'KXTS
159 NEXT I'BBCJ
160 IF F5=1 THEN 180'DGPE
170 J=42'BDWD
171 FOR K=1 TO 7:PRINT#4,H$;:NEXT K
    :PRINT#4'GNLJ
173 FOR K=6 TO 0 STEP -1
    :PRINT#4,C$(J-K);:NEXT K
    :PRINT#4,"[CMDR G]"'JTFP
175 FOR K=1 TO 7:FOR L=1 TO 7
    :PRINT#4,C$(0);:NEXT L
    :PRINT#4,"[CMDR G]":NEXT K'KXNS
180 FOR K=1 TO 7:PRINT#4,H$;:NEXT K
    :PRINT#4'GNLJ
190 REM CLOSE PRINTER CHANNEL AND
    QUIT'BBFM
192 CLOSE 4:END'CCQH
200 REM DATA'BESW
210 DATA "[CMDR G] 1","[CMDR G] 2","
    [CMDR G] 3","[CMDR G] 4","[CMDR G]
     5","[CMDR G] 6","[CMDR G] 7","
    [CMDR G] 8","[CMDR G] 9","[CMDR G]
    10"'BJSI
220 DATA "[CMDR G]11","[CMDR G]12","
    [CMDR G]13","[CMDR G]14","[CMDR G]
    15","[CMDR G]16","[CMDR G]17","
    [CMDR G]18","[CMDR G]19","[CMDR G]
    20"'BJAK
230 DATA "[CMDR G]21","[CMDR G]22","
    [CMDR G]23","[CMDR G]24","[CMDR G]
    25","[CMDR G]26","[CMDR G]27","
    [CMDR G]28","[CMDR G]29","[CMDR G]
    30","[CMDR G]31"'BKQM
240 DATA "JANUARY","FEBRUARY","MARCH",
    "APRIL"'BDYI
250 DATA "MAY","JUNE","JULY","AUGUST",
    "SEPTEMBER"'BEQK
260 DATA "OCTOBER","NOVEMBER",
    "DECEMBER"'BCXJ
270 DATA 31,28,31,30,31,30,31,31,30,
    31,30,31'BKZK
280 DATA "[SPACE3]SUNDAY[SPACE2]","
    [SPACE3]MONDAY[SPACE2]","[SPACE2]
    TUESDAY[SPACE2]"," WEDNESDAY
    "'BDEP
290 DATA "[SPACE2]THURSDAY ","[SPACE3]
    FRIDAY[SPACE2]","[SPACE2]
    SATURDAY "'BCPN
```

END



Damage Control reports a hit on the starboard side. Send in Alpha, Baker and Charlie to repair.

The time: 1400 hours. Somewhere in the Pacific. Some ill-fated coordinates in World War II.

eyeball to eyeball action. This time around you'll be right in the middle of it all. You knew it wouldn't be pretty. But how tough could it be to rescue a downed pilot?

Will it be the twin 40mm Bofors anti-aircraft guns? Or the 5″ lead-spewers aft? Depth charges or torpedoes? Autopilot or guts?

LIFE, LIBERTY AND THE HAPPINESS OF PURSUIT.



Radar spots inbound Zeroes. Ready forward gunnery positions. Man the anti-aircraft turrets. They're coming.

You're at the helm, commanding the greatest concentration of firepower ever put in a lightweight fighter. The deadly Fletcher Class Destroyer.

You've embarked on the first simulation that actually combines the intricate, large-scale strategy of wargaming with the intensity of furious,

It sure seemed a lot easier than shelling islands, escorting a convoy or hunting subs.

Or so you thought. But now look what you've got. Thirteen fully-operational, ear-bursting battle stations to worry about, all armed to the gills. Not to mention radar. Navigation. Sonar.

And half the Japanese fleet crawling up your spine.

Time to make some tactical decisions.

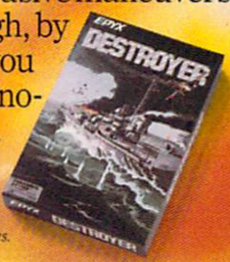
YOU'VE ALWAYS DREAMED OF BEAUTIFUL MODELS. Now you can pick up an entire assault fleet, including a replica Fletcher Class Destroyer, from Revell. Or win an authentic scrambled eggs *flight deck cap. Sweepstakes details are in every box, or write for an entry coupon. No purchase necessary. Sweepstakes ends June 15, 1987. Official rules are available at participating dealers.*

Any choice could be your last, so make it good.

Suddenly, you hear the ominous rumble of incoming Zeroes. You fire, and send one plummeting to the sea, trailing a plume of smoke.

On instinct, you instruct the bridge to commence evasive maneuvers.

Even though, by experience, you know there's nowhere to run.



EPYX®

Apple II & compatibles, C64/128, IBM & compatibles.

BY BRUCE JAEGER

Telephone Lister

for the PET, VIC 20, Plus/4, Commodore 16, 64 and 128

Sort and print your most frequently used telephone numbers.

If you're like me, you probably have a scribbled list of telephone numbers near your telephone, a sheet of paper that may have started out neatly, but soon ended up with additions penciled in every which way and old numbers scratched out and the new ones squeezed in—not always legibly. It finally dawned on me that this was a perfect job for a home computer!

Telephone Lister is a useful little program that will keep track of those most frequently used telephone numbers and reprint a sorted list of them for you whenever the old one is outdated. It will even resave itself automatically.

Type in the program carefully, substituting your own information for the phone names and numbers I give as examples in the DATA statements. Note that the name and numbers are separated by a slash. If there is a business number for a particular person, just add another slash and then that number.

The quotation marks surrounding each piece of data are there so we may use upper- and lower-case letters and commas in our data. As an experiment, try leaving out that first quotation mark, and then list that line number. The last DATA must be end or the program will give you an out of data error.

If you don't want to save the program under the name Phone Numbers as I did, change the name in lines 640 and 660 to whatever you want. And if you think you'll be using more than 50 names, just change the 50 in the DIM statement in line 140 to whatever you'll need.

The program should work unmodified on just about every Commodore home computer and with all of the Commodore printers, even my ancient model 2023. You can, of course, modify your own version of the program for screen colors and things like that.

No more sloppy lists! ■

Before typing these programs, read "How to Enter Programs," and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Telephone Lister

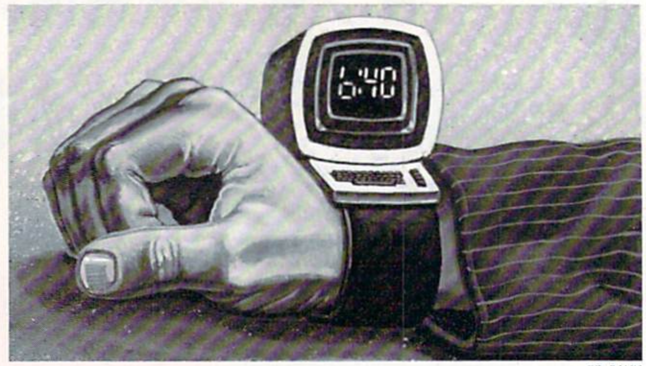
```
100 PRINT CHR$(147);CHR$(14);'DLNY
110 FOR X = 1 TO 40'DEUX
120 SP$=SP$+" " : NEXT'DHWA
130 PRINT "[SHFT R]EADING
    NUMBERS..."'BAID
140 DIM N$(50) : KT=0'CKEC
150 READ N$'BCXA
160 IF N$ = "END" THEN 240 'DFWE
170 FOR X = KT TO 1 STEP-1'FFGG
180 IF N$ >= N$(X) THEN 210 'EKFH
190 N$(X+1)=N$(X)'CLTH
200 NEXT X'BBRV
210 N$(X+1)=N$'CIAA
220 KT=KT+1'CFCA
230 GOTO 150'BDFA
240 REM ALL READ & SORTED'BOWE
250 PRINT : PRINT"[SHFT P]
    RINTER READY?"'CBPG
260 GET A$ : IF A$<>"Y" THEN 260 'FITH
270 LC$=CHR$(17) : REM MAKES OLDER
    CBM PRINTERS USE LOWERCASE'DQLQ
280 OPEN 4,4'BDAF
290 PRINT#4 : PRINT#4'CDJG
300 PRINT#4,LC$;"[SHFT T,SHFT E,
    SHFT L,SHFT E,SHFT P,SHFT H,
    SHFT O,SHFT N,SHFT E] [SHFT N,
    SHFT U,SHFT M,SHFT B,SHFT E,
    SHFT R,SHFT S]"'BGNL
310 PRINT#4'BBDX
320 PRINT#4,LC$;"[SHFT N,SHFT A,
    SHFT M,SHFT E]";'BHTE
330 PRINT#4,LEFT$(SP$,36);'CLRD
340 PRINT#4,LC$;"[SHFT H,SHFT O,
    SHFT M,SHFT E]";'BHCG
350 PRINT#4,LEFT$(SP$,14);'CLNF
360 PRINT#4,LC$;"[SHFT W,SHFT O,
    SHFT R,SHFT K]"'BGTI
370 FOR X = 1 TO 80'DEYG
380 PRINT#4,"-";'BDVG
390 NEXT : PRINT#4'CCKH
400 REM NOW PRINT NAMES & NUMBERS'BVWE
410 FOR X = 1 TO KT'DEEB
420 REM SEPARATE NAMES & NUMBERS'BVDG
430 N$ = N$(X)'BHMD
440 Y=1'BCJD
450 IF Y>LEN(N$) THEN 470 'EIRH
460 IF MID$(N$,Y,1)<>"/"THEN Y=Y+1
    :GOTO 450'IPHM
470 NA$ = LEFT$(N$,Y-1)'DKFJ
480 N$ = MID$(N$,Y+1)'DJRK
490 Y=1'BCJI
500 IF Y>LEN(N$) THEN HN$=N$ : BN$=""
    : GOTO 540 'HSFH
510 IF MID$(N$,Y,1)<>"/"THEN Y=Y+1
    :GOTO 500'IPDI
520 HN$ = LEFT$(N$,Y-1)'DKMF
530 BN$ = MID$(N$,Y+1)'DKHG
540 NA$=LEFT$(NA$+SP$,40)'DOYI
550 HN$=LEFT$(HN$+SP$,18)'DOSJ
560 PRINT#4,LC$;NA$;HN$;BN$'BRYJ
570 NEXT X'BBRG
580 PRINT#4 : CLOSE 4'CDRI
590 PRINT : PRINT'CBHJ
600 PRINT"[SHFT W]ANT TO RESAVE THIS
    VERSION ON DISK?"'BAEK
610 GET A$ : IF A$<>"Y"AND A$<>"N"
    THEN 610 'IKRI
620 IF A$="N" THEN END'ECYF
```

Continued on pg. 58

BY BENNETT COOKSON, JR.

The Clock

for the Commodore 64



ED SAUK

Put a digital clock anywhere on your screen.

The Clock is an interrupt-driven machine-language routine that adds a digital clock anywhere on your screen. It keeps on ticking even when other programs are running. The clock can be easily set and read from BASIC and is ideal as a timer in your own video game. A menu allows you to choose the color of your clock, position on screen and type of display. For example, you can choose a full clock or just a seconds timer.

The 6526 CIA chip that is used for I/O in the Commodore 64 has a Time of Day Clock. This clock keeps track of time in hours, minutes, seconds and tenths of seconds, and it even has an AM/PM flag. The Time of Day Clock is not the same as TI$. In fact it is not used by BASIC. The Clock is more accurate than TI$ and nothing but turning off the computer will upset the time.

How can this great little clock be put to use? How about if we wrote a BASIC program to peek at the clock and print it on the screen? However, it would be much better if the computer would print the clock automatically. That way other programs could be running at the same time.

An interrupt-driven routine is the answer. Sixty times a second the computer takes time out to read the keyboard and other housekeeping, called an interrupt. A program that runs at these interrupts is interrupt-driven.

The clock is a machine-language routine that reads the Time of Day Clock at every interrupt and pokes it on-screen. The clock on the screen is updated automatically 60 times a second.

The program is written as a BASIC loader. The BASIC loader has many extra features. It will, in fact, make changes to the machine-language program, customizing The Clock to fit your needs.

First, you must set the time. Just choose Set Time from the menu. When asked for hours and minutes, type in the correct time or press RETURN to leave it the same. The clock will not start until the time is set.

Next, you can move the clock around the screen or change the color. The best way to find out what you can do is to try everything on the menu. You can't hurt anything.

The clock defaults to displaying AM/PM, hours, minutes, seconds and tenths of seconds. A hidden colon is the first character. (It is hidden because it has no color. The colon separates the clock from commands typed on the same line.) All this will change when you choose Adjust Type from the menu. Keep pressing the same key to see all types. The machine-language program is actually being changed to jump over certain parts.

For example, you could put a seconds and tenths of seconds timer on your own game screen. The time could be easily set and read from BASIC and you would never have to print it to the screen so you can concentrate on the rest of the program.

Four functions have been defined to make reading and setting the clock easy. In addition, the variables HRS, MIN, SEC and TEN have been set to the location of the hour, minute, second and tenths of second registers. Thus, hour = FNR(HRS) will read the hours and FNR(MIN) will read the minutes. POKE HRS,FNS(hour) and POKE MIN,FNS(minute) will set hours and minutes.

The AM/PM flag is in the hours register and it defaults to AM when hours are set. POKE HRS,FNPM(0) will set the PM flag without disturbing the hours. POKE HRS,FNR(HRS) will set it to AM. FNRP(0) will read AM/PM flag, zero if AM 128 if PM. Lines 200-240 give good examples in reading and setting the clock.

Any time the hours are set or read, the clock stops (though it keeps time internally) until the tenths of seconds are set or read. This is so the clock won't change while it is being read.

The BASIC loader will also relocate the 116 bytes of machine language to run anywhere in memory. Change the start address (SA) in line 60 and the loader will do the rest. (Notice how everything in the program is referenced by SA.). The Clock does not use any zero page locations.

If you have a machine-language monitor, you can save the clock as machine language rather than a BASIC program. Use SA [49152,$C000] as the start address and SA + 116 [49268,$C075] as the end address. SYS(SA + 104) [49256] will start printing the clock to screen.

The advantage of saving the clock as machine language is that it can then be directly loaded into memory without the BASIC loader. Also, any adjustments you have made (color, position) will stay as they are when you save the machine language.

The Clock uses the interrupt routine vectors 788-789. SYS (SA + 104) executes a routine that changes these vectors, which starts the program running. Watch out for other programs that change these vectors because that will upset the clock. For example, POKE 788,52 is a common way to disable the RUN/STOP key, but it also causes the clock to jump all over the screen. To fix it, simply restart the clock, SYS(SA + 104), or press STOP/RESTORE to get it out of the way.

One of the real good things about the Commodore 64's Time of Day Clock is that it never loses time even when the interrupts are turned off—it just keeps on ticking.

The Program

Lines	
10-15	Calculate high and low bytes of a number.

Continued on pg. 117

Supercharge your Commodore for just $9.95 with LOADSTAR ...

... The Commodore Magazine On Disk



LOADSTAR 34 features these Special Issue Programs:

SDI Star Wars
Calendar Maker
Telephone Lister
Printer Maze 128
Inter Melody
Magic Music Box
Basic Macro Processor
The Clock
Instant Load and Directory
Program Sentry
Upgrading from 64 to 128
Beginner's Guide to Burst Mode
Interfacing Commodore's User Port
Falcon and the Enemy
The Moonlight Zone
Smokebusters 128
Hi-Res Helper
Permutations
Mirror

... PLUS these exclusive LOADSTAR features:

A Peek at Dispute?'s Gazelle
Pixal Puzzler

Each issue of LOADSTAR also contains music graphics, educational programs, applications, programming tips, product reviews ... and more!

THE COMPLETE FAMILY COMPUTER MAGAZINE ON DISK

LOADSTAR eliminates time-consuming typing and keystroke error. Each disk issue contains all the programs from Commodore Magazine in ready-to--run form, as if you typed them yourself.

In addition, LOADSTAR subscribers get programs that do not appear in the paper magazines. Each issue contains educational programs, useful applications, helpful programming tools, lively music, colorful animated graphics, and games and entertainment the whole family will enjoy.

LOADSTAR programs are unprotected for your convenience ... an added benefit for Commodore users who want to get the most from their C-64 or C-128 investment.

START YOUR SUBSCRIPTION TODAY!

☐ **$9.95 - LOADSTAR issue 34 featured above**
☐ **$9.95 - LOADSTAR issue featuring programs from Commodore MICROCOMPUTERS Mo____ Yr___**
☐ **$9.95 - LOADSTAR issue featuring programs from Commodore POWER/PLAY Mo____ Yr___**

*NAME*______________________
*ADDRESS*______________________
*CITY*______________ *STATE*______ *ZIP*___
*CARD #*______________________ *EXP*___

OR SAVE UP TO 50% WHEN YOU SUBSCRIBE!

☐ **$ 39.95 - 6 Months**
☐ **$ 69.95 - 12 Months**
☐ **$119.95 - 24 Months**

☐ *PAYMENT ENCLOSED*
☐ *VISA*
☐ *MASTERCARD*
☐ *AMERICAN EXPRESS*

THESE RATES ALSO APPLY IN CANADA AND MEXICO (U.S. FUNDS ONLY) WRITE FOR OVERSEAS RATES

CALL TOLL-FREE FOR IMMEDIATE SHIPMENT OF CREDIT CARD ORDERS

1-800-831-2694

(IN LOUISIANA CALL 1-318-868-7247)

To order by mail send to: LOADSTAR / P.O. BOX 30008 / SHREVEPORT, LA 71130-0008

BY ROBERT L. LYKINS

SDI Star Wars

for the Commodore 128



ED SAUK

The President's Strategic Defense Initiative has been realized. The future is now. But the star wars tracking computer is down for repairs and the backup has not yet been installed. An alarm sounds. Incoming enemy missiles are headed for major U.S. cities and you must manually aim the particle beam weapon to place energized fields in the missiles' paths.

The program is listed in two parts. The first is the main BASIC program consisting of some fairly lengthy lines. BASIC 7.0 allows a line length of 160 characters, twice as many as BASIC 2.0 permits. The lines are numbered with an increment of 5, allowing you to utilize the automatic line numbering feature built into the 128. Just type AUTO 5 [RETURN] and the beginning line number (100) with the first program line. When you hit RETURN, the next line number is displayed with the cursor to the right of it, ready for you to type the rest of the line.

The second part of the program is sprite data. I could have put it in DATA statements in the BASIC program, but it is easier to enter the data directly into memory using the 128's built-in machine-language monitor. Type MONITOR [RETURN] or hit F8 (if you have not changed the functions with the KEY command) to enter the monitor. Then type > followed by the hexadecimal address and data bytes separated by spaces.

If you are using 80 columns, you should type the whole line of 16 bytes before hitting RETURN. If you are in 40 columns, you should type only the first eight and hit RETURN. Then add 8 to the previous starting address and enter the next eight bytes. For example,

```
> 0E00 00 00 00 00 00 00 00 00
> 0E08 00 00 00 00 00 7F 00 01
> 0E10 80 C0 02 08 20 04 08 10
> 0E18 04 08 10 08 00 08 09 E3
```

When you finish, type X to return to BASIC, and save the data to disk with the BSAVE command:

BSAVE "SWS",B0,P3584 TO P3968

There is no need to save the whole area of sprite memory because the data is for only six sprites. The area saved takes only two blocks on disk and will load faster (at the start of the program) than would data for eight sprites.

Once you have saved the BASIC program and sprite data to disk, you are ready to play the game. Be sure your joystick is plugged into port 2. Load the program if it is not still in memory and run it. It should load the sprite data from disk, display the title screen, and play "The Washington Post March" while waiting for you to select instructions or one of three starting levels. View the instructions and pick Cadet if you are new to game playing. The Ensign level is best for experienced gamesters, but if you are bold and want to rack up points faster, pick Captain. The music ends and the game begins.

Your cities and defense post appear on the screen. A siren sounds and your gunsight centers on the screen. Eleven missiles start their descent at a speed dependent upon the level selected, leaving trails that will help you determine their trajectories. Move the gunsight to a point below a missile and press the firebutton to set up an energized field you hope the missile will enter. If all goes well, there will be an explosion, the missile will be destroyed, and its trail will disappear. Do the same with the other missiles to finish the wave.

You begin with 30 shots, five for each city you have standing. There is, however, a maximum of two energized fields at one time. If you fire a third shot before the first two fields dissipate, the first of the two will disappear to allow a field for the third shot. Therefore, you must not shoot too often or the missiles will not have time to enter a field before the field is gone. Your shot count is displayed prominently in the center of the scoreboard at the top of the screen.

You will notice immediately that the gunsight floats away in the direction you start it unless you redirect it with another push on the joystick. This enables it to move smoothly around the screen with occasional joystick manipulation. It will float off-screen if you allow it but will return on the opposite side. Do not press the firebutton when the gunsight is gone or you will transfer an energized field to its location. A beep will sound and you won't lose a shot.

You may transfer fields to screen extremities in this manner and they will continue to glow for the time they have remaining. This is handy if you need a field below the top of a city, a zone forbidden to a new shot. Please note that an energized field is ineffective if it touches the defense post. The speed of the gunsight gradually increases with higher levels and its movement may be hard to control without practice.

A city is devastated by a good hit from a missile. One such hit will turn the city red. Protect this city at all costs for if it is hit again before renovation it will burn to the ground and be lost forever. At each increase in level, the red city (if any) nearest the defense post is renovated (changes back to blue). The level increases every 10,000 or so points, beginning at 10,000 points when you select Cadet level, 20,000 points at Ensign level, and 30,000 points at Captain level. The game ends when all the cities have burned to the ground.

You receive 100 points multiplied by the play level for each missile destroyed. At the end of a wave, you receive a bonus of 100 points for each city left standing and 10 points for any shots remaining. Each hit on the defense post will halve the remaining shots. At the start of each wave you receive five shots per standing city. Near the end of the game you won't make a lot of bonus points from remaining shots.

Your present score is displayed on the left of the scoreboard. High score is displayed on the right. It may take several games for you to get a good idea of your score since the missile angles are selected randomly. Some games are harder than others. The highest I've scored after

Continued on pg. 36


TO THE VICTOR GO THE SPOILS
Actual Atari ST screen!
Actual Apple IIGS screen!
"What an experience!
Like being in a movie!"
Computer Entertainer, Dec. 1986
Actual Macintosh screen!
Actual Commodore 64 screen!
CINEMAWARE™
PRESENTS
DEFENDER OF THE CROWN
NOW PLAYING AT A SOFTWARE DEALER NEAR YOU
Exclusively distributed by Mindscape. 3444 Dundee Road Northbrook, IL 60062
Call Toll Free in Continental U.S. (Except IL): 1-800-443-7982 Illinois: 1-312-480-7667
COMMODORE, APPLE, MACINTOSH, ATARI AND IBM ARE REGISTERED TRADEMARKS OF
COMMODORE ELECTRONICS, LTD., APPLE COMPUTER INC., ATARI INC. AND INTERNATIONAL BUSINESS MACHINES INC. RESPECTIVELY
CINEMAWARE IS A TRADEMARK OF MASTER DESIGNER SOFTWARE, INC.

Continued from pg. 34

a few hours of practice is 135,310.

Game parameters, title screen and play screen are set up with the subroutine beginning at line 330. The TRAP statement sends the program to line 410 if there is an error. The most likely time this will happen other than your pressing the RUN/STOP key is a failed attempt at loading the sprite data with the BLOAD command, probably when the drive is not ready or the SWS file cannot be found. Syntax errors in the program could also be caught by this TRAP statement if you are careless in typing.

The third sprite provides the information for the cities and is transferred to B$ with the SPRSAV command. Six cities are placed on the screen with the loop in line 335 which utilizes the GSHAPE command in line 395. An offset of 80 pixels in the horizontal placement is added to cities four through six so there will be room for the defense post.

The SDI title is drawn using the CIRCLE command repeatedly for parts of the letters. Many shapes are possible by altering the various parameters of this command. Please note that variables A-F are used for the numbers 1-6 throughout the program. This was done to speed program actions since variables in program lines are acted upon faster than are numerals. Keep this in mind as you look over the program and it will help you understand what is happening.

The title screen music is played within the DO-WHILE loop starting in line 555 and ending in line 570. This loop also checks for level selection and flashes the selection numbers on and off.

The main game loop is between lines 105 and 145. Within this loop is a FOR-NEXT missile loop that positions sprite 5 (E) at the front of each missile trail and draws a line from the top of the screen to that position. Line 110 checks for a bump between this sprite and energized field one (sprite B). This is done again in line 120 after sprite B is moved to the spot of the second field.

The BUMP(1) function is rather tricky. It must be cleared between bumps by another reference to it and a period of time must go by before it is checked again. Sprite to sprite collisions are checked rather than sprite to graphic screen data collisions (BUMP(2)) so that stars could fill the sky. It wouldn't do to have an energized field colliding with a star.

The WAIT command in line 120 permits you to pause the game by depressing the SHIFT LOCK key. Press it again to resume play.

The joystick is checked twice per missile loop (lines 115 and 130) to increase response. Lines 150-185 determine the direction and speed of sprite 1, the gunsight. If the fire button is pressed and you have remaining shots, the program will jump to the fire routine at line 210. This routine draws and erases a fire line to the gunsight position and places sprite 2, the energized field, at this spot.

If a bump is detected between a missile and an energized field, the explosion routine, lines 195-205, is implemented, decrementing the missile count (G) and incrementing the score (SC). Back in the main loop, if the missile count becomes zero before the missiles drop to the limit (Y<1), the program jumps to the tally routine at line 275. Otherwise, the missiles keep dropping until they reach the limit and position checks are made to see if they have hit cities or the defense post. If so, appropriate action is taken in lines 230-270, before the tally.

At the end of tally, if you have no cities remaining, you get no shots for another wave and it is thus determined that the game is over. The program then jumps to the end of game routine at line 420. High score is adjusted, if necessary, a sad tune is played (line 450), and you are given the option of playing again or quitting.

A lot of planning is required to make a game programmed in BASIC fast enough to be enjoyable. Use of variables for numbers that are repeatedly used will speed the game, as will the use of a period in place of a lone zero. Rather than check within the main loop for each possibility of the joystick (IF JOY(2)=1), it is better to check for its being true (not zero) and then, if so, to use the ON-GOSUB command to take action. Replacing the AND logical operator with THEN-IF will also increase speed. An example, IF A=B AND C=D THEN... will be faster if changed to IF A=B THEN IF C=D THEN....

I have tried here to explain the logic of the game rather than the details of BASIC commands and statements. The user's manual provides detailed descriptions of the use of BASIC 7.0 keywords and other articles have discussed them time and again. An examination of the program after playing it a few times should give you a good idea of how the commands, statements and functions work. ■

Sprite Data

```
0E00  00 00 00 00 00 00 00 00  00 00 00 00 00 7F 00 01
0E10  80 C0 02 08 20 04 08 10  04 08 10 08 00 08 09 E3
0E20  C8 08 00 08 04 08 10 04  08 10 02 08 20 01 80 C0
0E30  00 7F 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0E40  00 00 00 20 82 08 08 28  20 3E 14 BC 0C BE 30 0B
0E50  69 E0 86 D7 92 2D BE 78  3B 69 EC 26 D7 98 95 AA
0E60  56 26 D7 98 3B 69 EC 2D  BE 78 86 D7 92 0B 69 E0
0E70  0C BE 30 3E 14 BC 08 28  20 20 82 08 00 00 00 00
0E80  30 00 12 FC 01 3F AC 51  00 FC 73 BF 84 73 80 FC
0E90  73 BF 94 7F 80 FC C3 BF  B4 3C 80 FC C3 BF 84 3C
0EA0  80 FC C3 BF D4 3C 80 FC  C3 BF A7 3C 80 FF C3 BF
0EB0  D7 3C 80 FF C3 BF B7 3C  80 FF C3 BF B7 FF F3 FF
0EC0  A0 08 02 64 0C 26 E6 46  3E 16 96 68 47 50 EE 4E
0ED0  DA 46 6E CB 14 68 93 88  71 3B 8C 73 3D 8F 8E FD
0EE0  C6 9C B7 0A C9 24 72 DB  9A A6 21 C7 76 40 E9 08
0EF0  74 F7 26 76 0A B5 C7 5B  3D 53 F9 C2 E6 9F 3D 00
0F00  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0F10  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0F20  00 00 10 00 00 00 00 00  00 00 00 00 00 00 00 00
0F30  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0F40  0F 99 F0 1F 18 F8 3F 18  FC 7E 18 7E FE 24 7F FF
0F50  FF FF 80 66 01 FF FF FF  81 E7 81 FF FF FF 87 E7
0F60  E1 FF FF FF 9F E7 F9 FE  00 7F B7 E7 ED DB 7E DB
0F70  ED BD B7 B6 DB 6D ED BD  B7 92 42 49 FF FF FF FF
```

Before typing this program, read "How to Enter Programs" and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

BASIC Program

```
100 GOSUB 330'BDJV
105 Y=Y-U*3:COLOR A,B:FOR L=.TO 10
    :IF G(L)THEN RX(L)=RX(L)+RB(L)
    :MOVSPR E,RA(L)+RX(L)+13,180-Y+40
    :DRAW,RA(L),25 TO RA(L)+RX(L),
    180-Y'UGLH
110 IF BUMP(A)=18 THEN IF G(L)THEN
    GOSUB 190'HNLC
115 J=JOY(B):MOVSPR B,SX,SY:K=BUMP(A)
    :ON J-(J AND 128)GOSUB 150,155,
    160,165,170,175,180,185
    :IF J AND 128 THEN IF AR THEN
    GOSUB 210'PSPA
120 WAIT 211,A,A:IF BUMP(A)AND 18
    THEN IF G(L)THEN GOSUB 190'IVKG
125 IF S THEN SPRCOLOR RND(A)*15+B,
```

Continued on pg. 38

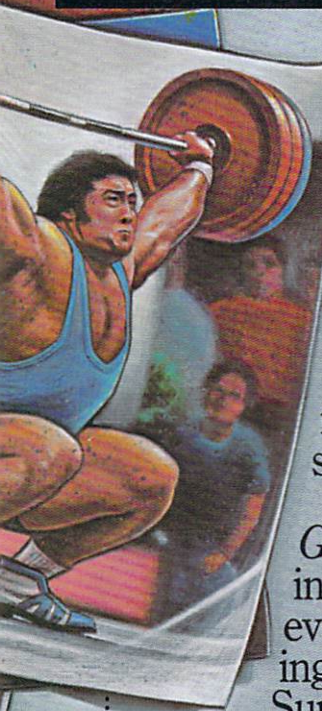


STAY ON TOP OF WORLD EVENTS.

The following programs are brought to you by an incredible series of events.

Over 30 of them, to be specific.

They're the unforgettable events of our best-selling Games series.



First, *World Games.*™ Eight international events ranging from Sumo wrestling to cliff diving. Bull riding to weightlifting. Even skiing the Chamonix slalom.

There's our enormously popular *Summer Games.*® Break records in track, swimming, diving, gymnastics and skeet shooting, just to name a few.



Sumo wrestling. A sport of ritual and tradition.



Gymnastics. A graceful display of balance, poise and concentration.



The Bobsled. One wrong move and it's right down the tube.

There's its equally acclaimed sequel, *Summer Games II.*™ Go for the gold in rowing, cycling, equestrian, fencing, the high jump, the triple jump, kayaking, and more.

Finally, there's the icy challenges of *Winter Games.*™ Wait'll you careen the tube in a bobsled. Fly the ski jump. Or choreograph an entire figure skating routine. You'll flip out over the hot dog aerials. And beg for mercy in the Biathlon. And coming next summer, our newest Games program. (Better get ready to hit the beach.)



Cliff Diving. In Acapulco, everyone falls for it.



Try your hand at Fencing.



Ride a bucking bull.

Now, it may seem like a lot to handle.

But don't let world events get the best of you.





1 To 8 players. Amiga, Apple II & compatibles, Atari ST, C64/128, IBM & compatibles. Coming for the Apple IIGS.

1 To 8 players. Apple II & compatibles, Atari, C64/128.

1 To 8 players. Apple II & compatibles, C64/128, IBM & compatibles.

1 To 8 players. Amiga, Apple II & compatibles, Atari ST, C64/128, IBM & compatibles, Macintosh.

EPYX

Continued from pg. 36

```
    RND(A)*15+B:MOVSPR B,PX,PY
    :ELSE SPRITE B,.:SX=344'NKMT
130 J=JOY(B):ON J-(J AND 128)GOSUB
    150,155,160,165,170,175,180,185
    :IF J AND 128 THEN IF AR THEN
    GOSUB 210'MDUQ
135 NEXT:IF Y<A THEN 230'EGHG
140 IF S THEN S=S-A:SPRITE B,,
    RND(A)*15+B'IPLH
145 IF G THEN 105:ELSE 275'DIMH
150 MOVSPR A,.#T:RETURN'CHVC
155 MOVSPR A,45#T:RETURN'CIZI
160 MOVSPR A,90#T:RETURN'CIZE
165 MOVSPR A,135#T:RETURN'CJWJ
170 MOVSPR A,180#T:RETURN'CJWF
175 MOVSPR A,225#T:RETURN'CJWK
180 MOVSPR A,270#T:RETURN'CJWG
185 MOVSPR A,315#T:RETURN'CJWL
190 IF BUMP(A)=18 THEN 195
    :ELSE RETURN'GKKJ
195 MOVSPR B,-12,-9:SPRITE B,,,,A,A
    :G(L)=.:SOUND C,3000,30,B,300,300,
    C:COLOR D,E:DRAW.,RA(L),
    25 TO RA(L)+RX(L),180-Y:COLOR D,8
    :SC=SC+100*U:G=G-A'RRQQ
200 COLOR D,F:SPRITE B,,,,.,.
    :MOVSPR B,+12,+9'FVEE
205 COLOR A,8:CHAR A,13-LEN(STR$(SC)),
    A,STR$(SC):COLOR A,B:RETURN'IBNM
210 IF S THEN SX=PX:SY=PY'EKCC
215 PX=RSPPOS(A,.):PY=RSPPOS(A,A)
    :IF PX>20 THEN IF PX<320 THEN IF
    PY<220 THEN IF PY>70 THEN 220
    :ELSE SOUND A,55000,E:RETURN'TAAA
220 SOUND B,30000,10,.,9999,999,C
    :DRAW,159,180 TO PX-12,PY-40
    :MOVSPR B,PX,PY:S=C:SPRITE B,A
    :AR=AR-A:DRAW.,159,180 TO PX-12,
    PY-40'OIXC
225 CHAR A,18,A,STR$(AR)+" "
    :RETURN'EMLI
230 FOR L=.TO 10:IF G(L)=.THEN
    270'GNWF
235 FOR M=A TO F:IF RA(L)+RX(L)>=XC(M)
    THEN IF RA(L)+RX(L)<=XC(M)+24
    THEN IF E(M)<>9 THEN
    E(M)=E(M)+B'UAJD
240 IF E(M)=B THEN E(M)=C
    :SOUND A,9999,70,B,999,999,.
    :FOR K=A TO 20:SPRITE C,,,A
    :FOR N=A TO C:MOVSPR C,XC(M)+22+N,
    229:NEXT:NEXT:MOVSPR C,.,.
    :COLOR A,9:GSHAPE B$,XC(M),
    179'UPKI
245 IF E(M)=9 THEN 255'DIFI
250 IF E(M)=E THEN MOVSPR D,XC(M)+24,
    229:SOUND C,5000,150,A,50,30,C
    :FOR K=A TO 85:SPRITE D,A,
    RND(A)*15+B,RND(.)*B:NEXT
    :SPRITE D,.:GSHAPE C$,XC(M),179
    :E(M)=9'TMEH
255 NEXT'BAEG
260 IF AR THEN IF RA(L)+RX(L)<184
    THEN IF RA(L)+RX(L)>136 THEN
    AR=INT(AR/B):SOUND A,400,99,,,,C
    :FOR K=.TO 99:SPRITE F,A,
    RND(A)*16+A:NEXT:COLOR A,B
    :GOSUB 225:SPRITE F,A,11'AIXM
265 DRAW.,RA(L),25 TO RA(L)+RX(L),
    180-Y'EASP
270 NEXT'BAED
275 IF AR THEN SOUND A,55000-1000*AR,C
    :AR=AR-A:SC=SC+10:GOSUB 225
    :GOSUB 205:GOTO 275'MRUA
280 SPRITE B,.:SPRITE F,.:FOR L=A TO F
    :GOSUB 400:IF E(L)=9 THEN 295'JBBP
285 MOVSPR C,XC(L)+24,229:SPRITE C,,,.
    :SC=SC+100:SOUND A,10000+L*500,20
    :SOUND B,12000+L*600,20'LJYE
290 GOSUB 205:AR=AR+E:GOSUB 225
    :COLOR A,15-B*E(L):GSHAPE B$,
    XC(L),179:MOVSPR C,.,.'JSUU
295 NEXT'BAEK
300 IF SC<U*10000 THEN 325'ELHC
305 FOR L=A TO 50:SOUND C,
    RND(A)*40000+999,A:G=RND(A)*16+A
    :COLOR A,G:COLOR D,G
    :CHAR A,13-LEN(STR$(SC)),A,
    STR$(SC):NEXT:GOSUB 205
    :GOSUB 225'VNGF
310 U=U+A:FOR L=D TO F:IF E(L)AND
    E(L)<9 THEN E(L)=.:GOSUB 405
    :GSHAPE B$,XC(L),179:L=F
    :GOTO 320'OVOR
315 IF E(7-L)AND E(7-L)<9 THEN
    E(7-L)=.:GOSUB 405:GSHAPE B$,
    XC(7-L),179:L=F:GOTO 320'NRVU
320 NEXT'BAEY
325 IF AR THEN GOSUB 355:GOTO 105
    :ELSE 420'FNLJ
330 TRAP 410:BLOAD"SWS":COLOR.,12
    :GRAPHIC 1,1:SPRSAV 3,B$:A=1:B=2
    :C=3:D=4:E=5:F=6:U=RND(-TI)'OTAV
335 COLOR A,15:COLOR D,E:K=.
    :FOR L=A TO F:MOVSPR L,.#.
    :MOVSPR L,L*50,.:IF L>C THEN K=80
    :GOSUB 395:NEXT:ELSE GOSUB 395
    :NEXT'SWSC
340 U=I:SSHAPE C$,30,30,54,54
    :FOR L=A TO 50:GOSUB 400:NEXT
    :AR=30:COLOR A,F:CHAR A,.,.,"
    [CMDR +4,SHFT POUND,CMDR U8,
    CMDR *,CMDR +4,SHFT POUND,CMDR U2,
    CMDR *,CMDR +4,SHFT POUND,CMDR U8,
    CMDR *,CMDR +4]"'KSFU
345 CHAR A,.,B,"[CMDR +4,CMDR *,
    CMDR U8,SHFT POUND,CMDR +4,CMDR *,
    CMDR U2,SHFT POUND,CMDR +4,CMDR *,
    CMDR U8,SHFT POUND,CMDR +4]",A
    :FOR L=A TO F:E(L)=.:NEXT:SC=.
    :SPRITE C,A,8,,.,.,.,.'IMGW
350 CHAR A,.,A,"[SHFT V4,SPACE8]0
    [SHFT V4] 30 [SHFT V4,SPACE8]0
```

Continued on pg. 42

COMMODORE

C-64-C COMPUTER

INCLUDES GEOS



NEW!

C-64-C

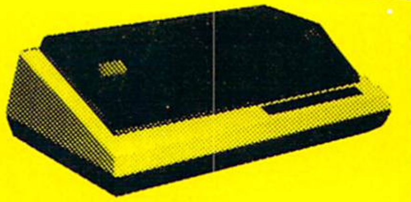
$168

*WITH PURCHASE OF DISK DRIVE

PRINTERS

803-C
COMMODORE

$139



star micronics inc



$198

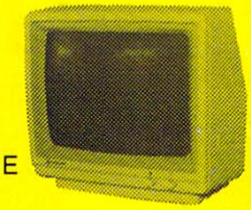
LETTER QUALITY DAISYWHEEL

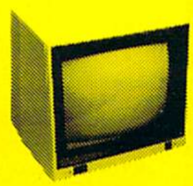
MONITORS

RETAIL $120

$59

MONOCROME MONITOR





RETAIL $299

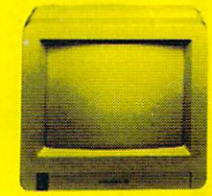
$139

FULL COLOR

RGB

RETAIL $400

$199



PRO-TECH-TRONICS

6860 Shingle Crk. Pkwy. #201
Minneapolis, MN 55430
(612) 560-6603

CALL TODAY!

QUICK-DELIVERY

VISA

OPEN 7 DAYS A WEEK & EVENINGS

1-800-345-5080

PRO-TECH TRONICS

C-64 COMPUTER



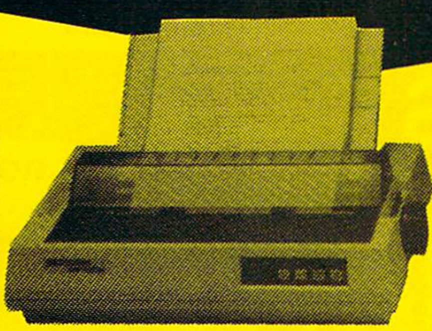
$99

*WITH PURCHASE OF SPECIALLY PRICED SOFTWARE PACKAGE

FACTORY SERVICED
FULL WARRANTY

FREE HOME TRIAL

PRINTER



$149

SEIKOSHA

NEAR LETTER QUALITY
- 100 CPS Draft Mode
- 20 CPS NEAR LETTER QUALITY MODE
- Commodore Ready

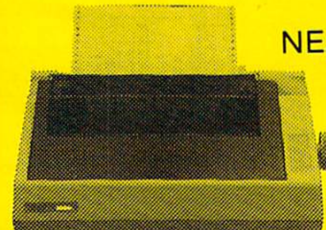
PRINTER

star micronics

SG-10-C

$198

NEAR LETTER QUALITY



COMMODORE READY

- 120 CPS Draft
- 30 CPS NEAR LETTER QUALITY
- Includes Both...
  —TRACTOR FEED
  —FRICTION

FREE TRIAL PERIOD

CALL TODAY
OPEN 7 DAYS
A WEEK &
EVENINGS

1-800-345-5080

Continued from pg. 38

```
    [SHFT V4]":IF H THEN COLOR A,D
    :CHAR A,35-LEN(STR$(H)),A,
    STR$(H)'JCYG
355 T=B+INT(U/C):SPRITE A,A,14,.,.,.,.
    :SPRITE B,.,,A,.,.,A:S=.:G=11'ISKV
360 MOVSPR F,160,229:SPRITE F,A,11,,A,
    .,.:PRINT CHR$(7);:IF RA(.)=.THEN
    475'HPDP
365 FOR L=.TO 10:RX(L)=.'ELTN
370 RA(L)=INT(RND(A)*318+A)
    :RB(L)=INT(RND(A)*(9*U))'KEPR
375 IF INT(RND(A)*2)THEN
    RB(L)=-RB(L)'HQIR
380 IF RA(L)+RB(L)*60/U<5 OR
    RA(L)+RB(L)*60/U>314 THEN 370'LIJT
385 G(L)=A:NEXT'CGVM
390 COLOR D,8:MOVSPR A,.#.
    :MOVSPR A,171,140:SPRITE A,A,14
    :SOUND A,25000,60,.,6000,999
    :SLEEP A:COLOR D,F:SPRITE E,A,12
    :Y=155:RETURN'KWKD
395 SOUND A,35000,20,B,5000,5000,.
    :XC(L)=(L-.5)*36+K:GSHAPE B$,
    XC(L),179:RETURN'HCDB
400 COLOR A,B+14*INT(RND(A)*B)
    :DRAW,RND(A)*320,RND(A)*150+25
    :RETURN'NEEM
405 COLOR A,15:SOUND A,35000,20,B,
    5000,5000,.:RETURN'DFAL
410 GRAPHIC.:IF ER<>30 THEN PRINT"
    [CLEAR,DOWN,WHITE]FIX AND HIT A
    KEY.[DOWN2]"ERR$(ER):GET KEY A$
    :ELSE END'LOQP
415 RUN'BAME
420 GRAPHIC C,.:SOUND B,600,220
    :SOUND C,8000,220,A,,30,C
    :COLOR D,C:FOR L=1 TO 150
    :COLOR.,RND(A)*16+A:VOL 15-L/10
    :NEXT:COLOR.,12:GRAPHIC A,.
    :COLOR D,7'SVVD
425 MOVSPR A,.#.:COLOR A,D
    :IF SC>H THEN H=SC:CHAR A,
    35-LEN(STR$(H)),A,STR$(H)'LICU
430 CHAR A,25,C,"[CMDR *,SPACE10,
    SHFT POUND]":COLOR A,8
    :CHAR A,C,C,"[CMDR *,SPACE10,
    SHFT POUND]"'DSON
435 CHAR A,D,C,"YOUR SCORE",A
    :COLOR A,D:CHAR A,26,C,
    "HIGH SCORE",A'DWBT
440 IF RA(.)=.THEN RETURN'EGCF
445 VOL 4:GOSUB 475:RESTORE
    :TEMPO 9'EJCL
450 READ D$:IF D$<>"*"THEN PLAY D$
    :GOTO 450'HMYK
455 COLOR A,14:CHAR A,C,19,
    " TO PLAY AGAIN, PRESS FIRE
    BUTTON ",A'COKV
460 COLOR A,11:CHAR A,7,21," TO QUIT,
     PRESS SPACE BAR ",A'COEP
465 GET A$:IF A$=" "THEN GRAPHIC.
    :END'GHKO
470 IF JOY(B)AND 128 THEN SCNCLR
    :GOSUB 335:GOTO 105
    :ELSE GOTO 465'JSIO
475 MOVSPR A,171,150:IF RA(.)=.THEN
    COLOR A,D:GOSUB 430'GYGT
480 COLOR A,8:CIRCLE,50,70,50,F,,,45,
    60:CIRCLE,49,70,50,F,,,45,60'DSYR
485 CIRCLE,84,50,E,10,,,,60
    :CIRCLE,15,90,E,10,,,,60'CKRT
490 CIRCLE,57,41,30,E,,,350,60
    :CIRCLE,42,99,30,E,,,350,60'CQDQ
495 CIRCLE,110,65,35,E,,,90,60
    :CIRCLE,145,65,25,E,,,90,60'CQHV
500 CIRCLE,128,42,15,D,,,12,60
    :CIRCLE,128,88,15,D,,,348,60'CRKI
505 CIRCLE,175,60,20,D,,,90,60'BUKI
510 CIRCLE,175,40,10,C,,,,60
    :CIRCLE,175,80,10,C,,,,60'CMPI
515 COLOR A,D:CHAR,28,8,"S"
    :CHAR,30,7,"T":CHAR,32,F,"A"
    :CHAR,34,E,"R"'FGYQ
520 CHAR,29,10,"W":CHAR,31,9,"A"
    :CHAR,33,8,"R":CHAR,35,7,"S"'EDLL
525 SOUND C,40000,30,,10000,1000,C
    :COLOR A,11:CIRCLE,200,145,180,50,
    270,32,,E'DFXT
530 COLOR A,B:DRAW,245,33 TO 195,75
    :COLOR A,8:CHAR A,12,16,
    "ROBERT L. LYKINS"'FFYQ
540 IF RA(.)THEN RETURN
    :ELSE RA(.)=1'FMNI
545 COLOR A,14:CHAR A,8,18," CHOOSE
    :[SPACE2]0 INSTRUCTIONS"
    :CHAR A,18,19,"-[SPACE2]-"
    :CIRCLE,155,155,C,B
    :CIRCLE,165,155,C,B,315,50
    :DRAW,162,153 TO 162,157'HUPK
550 COLOR A,14:CHAR A,.,20," LEVEL
    :[SPACE2]1 CADET[SPACE3]2 ENSIGN
    [SPACE3]3 CAPTAIN"'CMWR
555 RESTORE 700:TEMPO 15
    :DO WHILE A$=""'FKEO
560 GET A$:READ D$:IF D$="*"THEN
    RESTORE 700:GOTO 560'HPVM
565 PLAY D$:CHAR A,18,18,"0",N
    :CHAR A,9,20,"1",N:CHAR A,19,20,
    "2",N:CHAR A,30,20,"3",N'FVYY
570 N=ABS(N-A):LOOP:I=VAL(A$)
    :IF I>C THEN 555:ELSE IF I=.THEN
    580'NYBU
575 SCNCLR:GOTO 335'CETN
580 D$="THE STAR WARS DEFENSE
    TRACKING COMPUTER IS DOWN SO YOU
    MUST AIM THE PARTICLE BEAM TO
    DESTROY THE INCOMING
    MISSILES."'BCYO
582 D1$=" LOSE ALL YOUR CITIES AND
    IT'S OVER."'BDIT
585 A$="[SPACE40]":D$=A$+D$+D1$'EMLW
590 COLOR A,16:FOR L=A TO 152
    :CHAR A,.,18,MID$(D$,L,40)
```

```
    :FOR M=A TO 75:NEXT:NEXT'LJQW
595 SLEEP B:CHAR A,.,18,
    "THIS IS YOUR FLOATING GUN SIGHT.
    [SPACE8]"'CKMB
600 SOUND C,15000,30:COLOR A,B
    :CHAR A,23,13,"<-":SLEEP B
    :CHAR A,23,13,"[SPACE2]":SLEEP A
    :CHAR A,23,13,"<-":SLEEP B
    :CHAR A,23,13,"[SPACE2]"
    :COLOR A,16'KOXX
605 SLEEP B:CHAR A,.,18,
    "THIS IS AN INCOMING MISSILE
    TRAIL.[SPACE6]"'CKFS
610 SOUND C,15000,30:COLOR A,B
    :CHAR A,25,E,"->":SLEEP B
    :CHAR A,25,E,"[SPACE2]":SLEEP A
    :CHAR A,25,E,"->":SLEEP B
    :CHAR A,25,E,"[SPACE2]"
    :COLOR A,16'KKQX
615 SLEEP B:CHAR A,.,18,
    "THIS IS ONE OF SIX CITIES.
    PROTECT THEM."'CKIU
620 SOUND C,15000,30:COLOR A,B
    :CHAR A,.,23,"->":SLEEP B
    :CHAR A,.,23,"[SPACE2]":SLEEP A
    :CHAR A,.,23,"->":SLEEP B
    :CHAR A,.,23,"[SPACE2]"
    :COLOR A,16'KKQY
625 SLEEP B:CHAR A,.,18,
    "THIS IS THE DEFENSE POST YOU
    SHOOT FROM."'CKMV
630 SOUND C,15000,30:COLOR A,B
    :CHAR A,15,23,"->":SLEEP B
    :CHAR A,15,23,"[SPACE2]":SLEEP A
    :CHAR A,15,23,"->":SLEEP B
    :CHAR A,15,23,"[SPACE2]"
    :COLOR A,16'KOJB
635 D$="PLACE SIGHT BELOW MISSILE.
    PRESS FIRE BUTTON. ALLOW MISSILE
    TO HIT A RESULTING ENERGIZED
    FIELD TO DESTROY IT."'BCRN
637 D1$=" MAXIMUM OF TWO FIELDS AT
    ONCE. GET 'EM!!!"'BDCV
640 D$=A$+D$+D1$'DJLH
645 SLEEP B:FOR L=A TO 152
    :CHAR A,.,18,MID$(D$,L,40)
    :FOR M=A TO 75:NEXT:NEXT'LHFX
650 SLEEP B:CHAR A,.,18,A$:A$=""
    :GOTO 545'ETPL
655 DATA V1O3H.DV3O4Q$BV2RO3HFV3O4QAF
    'BDOR
660 DATA V1O3H.DV3O4HDV2QRO3HFV3O5QD'
    BCLN
665 DATA V1O3H.DV3O4$BV2QRO3HGV3O5QD'
    BCYS
670 DATA V1O3H.DV3O4.AV2QRHF'BTVM
675 DATA V1O3H.EV2QRV3O4SGAI$BV2O3H$B
    V3O5QDICO4$B'BPOW
680 DATA V1O3H.FV3O4AV2QRO4CV3O5QF'BA
    QP
685 DATA V1O3H.#CV2QRV3O4SEFIGV2O3HGV
    3O4QAA'BJHW
690 DATA V1O3H.DV2.FV3O4.D'BRQN
695 DATA*'BBWO
700 DATA V1T7O2QDV2T7O5DV3T7O6DV1O2I#
    DV2O5#DV3O6#DV1O2Q.EV2O5.EV3O6.E'
    BKGQ
705 DATA V1O2EV2O5EV3O6EV1O2I$EV2O5$E
    V3O6$EV1O2QDV2O5DV3O6DV1O2I#CV2O5
    #CV3O6#C'BTUX
710 DATA V1O2QDV2O5DV3O6DV1O2I#DV2O5#
    DV3O6#DV1O2Q.EV2O5.EV3O6.E'BEUQ
715 DATA V1O2EV2O5EV3O6EV1O2I$EV2O5$E
    V3O6$EV1O2QDV2O5DV3O6DV1O2I#CV2O5
    #CV3O6#C'BTUY
720 DATA V1O2QDV2O5DV3O6DV1O2I#DV2O5#
    DV3O6#DV1O2QEV2O5EV3O6EV1O2I$EV2O
    5$EV3O6$E'BUXV
725 DATA V1O2QDV2O5DV3O6DV1O2I#CV2O5#
    CV3O6#CV1O2QCV2O5CV3O6CV1O1IBV2O4
    BV3O5B'BRAA
730 DATA V1O1QAV2O4AV3O5AV1IRV2RV3RV1
    O2QDV2O5DV3O6DV1Q.RV2.RV3.R'BFCT
735 DATA V3O5IDD#CDM'BLMM
740 DATA V1O2ICV2O5CV3EV1RV2RV3RV1O3A
    V2O4CV3EV2O5Q.CV3.EV1O3IARA'BFYU
745 DATA O2DV2O5CV3#FRV2RV1RO3AV2O4CV
    3DO5Q.#FV2.CV1O3IARA'BXVW
750 DATA O2GV2O4BV3O5ARV2RV1RO3GV2BV3
    O4DO5Q.GV2O4.BV1O3IGRG'BAVT
755 DATA V2O4QBV3O5GV1O2IGRO3GV2O4DV3
    O5GGV2O4DV1O3GRV2RV3O5#FEV2O4DV1O
    3G'BNED
760 DATA O3#FV2O5CV3DRV2RV1RO4CV2O5CV
    3#DEV2CV1O3DRV2RV3RO5$EV2CV1O4C'B
    JCW
765 DATA V2O5QCV3DV1O3I#FRO4CV2O5CV3D
    DV2CV1O3DRV2O5CV3DDV2CV1O4C'BFFC
770 DATA O3GV2O4BV3O5DRV2RV1RO3BV2O4B
    V3O5#DEV2O4BV1O3DRV2RV3RO5$EV2O4B
    V1O3B'BQWA
775 DATA V2O4QBV3O5DV1O3IGRBV2O4DV3O5
    DDV2RV1DRV2RV3O5#CDV2O4DV1O3B'BHOD
780 DATA#CV2O4AV3O5ERV2RV1RO3GV2AV3RV
    2O4HAV3O5Q.EV1O3I#CRGCV3O5#F'BHDY
785 DATA RV2RV1RO3#FV2AV3RV2O4HAV3O5Q
    .#FV1O3ICR#FO2BV3O5A'BXBC
790 DATA RV2RV1RO3DV2GV3RO5Q.GV2O4.DV
    1O2IBRO3D'BMFU
795 DATA V3O5QGV2O4DV1O2I$BRO3DV2GV3O
    5GGV2RV1O2$BRV2RV3O5#FGV2O3GV1D'B
    JCF
800 DATA V3O5Q#FV2RV1O2IARO3DV2O4#FV3
    O5#F#FV2O4#FV1O2ARV2O4GV3O5G#GV2O
    4#GV1O3D'BTCU
805 DATA V3O5QAV2O4AV1O2IARO3EV2O4AV3
    O5AAV2O4AV1O2ARV2O4BV3O5BO6#CV2O5
    #CV1O3E'BSIA
810 DATA DV2O5DV3O6DRV2RV1RO2AV2O5GV3
    O6#CDV2O5#FV1O2DRV2RV3RO6#CV2O5GV
    1O2A'BPMU
815 DATA HDV2O5#FV3O6DQRM'BQWM
820 DATA*'BBWE
```

END

• Famous National Brand •

15 Day Free Trial-Satisfaction or Your Money Back

90 Day Immediate Replacement Warranty

Fantastic Price

BIG BLUE

Super Print Quality

80 Column Printer • 8½" Letter Size

We Like this Printer so much
We Bought Out the Factory

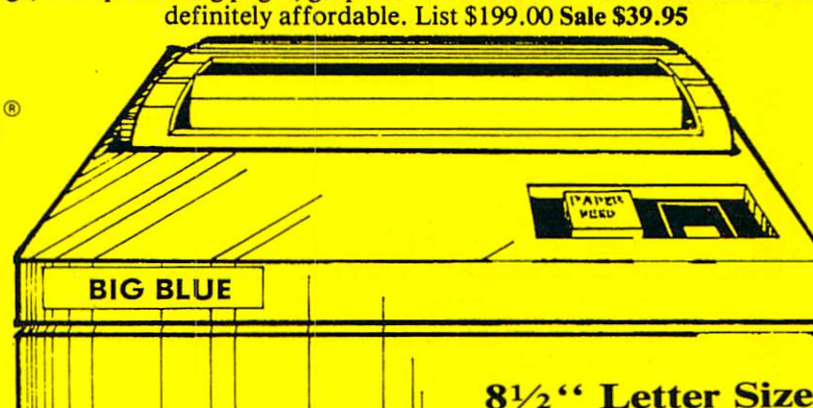
SALE $39.95 SALE

List $199.00

Now you can have a full fledged 8½" letter size 80 column printer for less than the cost of a large box of paper. This printer uses advanced dot matrix, heat transfer technology to print upper and lower case (with true lower descenders), underline, enlarged, all points addressable graphics plus More. Print out pictures, program listings, wordprocessing pages, graphics and more. Perfect for the homeowner or student and definitely affordable. List $199.00 **Sale $39.95**

Commodore® **Atari®**

IBM® **Apple®**



8½" Letter Size

This printer was made by **Canon®** for the IBM PC and PCjr. The Big Blue printer comes ready to hook up to the serial port of the **IBM®**PC jr. Plus with low cost adapter cables you can connect the Big Blue printer to the **Apple®** II,IIe,IIc, **Apple Compatibles, Atari®**, **Commodore®** 64 ,128, SX-64, Vic 20, Plus 4, **IBM®**PC, **IBM®**XT, **IBM®**AT, **IBM Compatibles**, TI-99, Tandy 1000, plus many more.

Intelligent Commodore® Interface—Connects Big Blue to the printer port of the C-64 and 128. Print graphics, us Printshop, word processors and more .... List $49.95 **Sale $19.95**

Intelligent Atari® Interface—Connects the Big Blue to Atari® computers (except 1200). Print Atari graphics, Printshop, word processors, and more .... List $49.95 **Sale $19.95**

RS-232 Adapter—Adapts Big Blue for IBM® PC, AT, XT, PCjr. & Apple® II series RS-232 port .... List $19.95 **Sale $ 9.95**

Laser128, Apple II®c Interface ... List $24.95 **Sale $12.95**

Printer Stand .... List $24.95 **Sale $14.95**

Paper (2 Rolls).......... List $19.95 **Sale $ 5.95**

Single Sheet Paper (Qty 500) .... List $29.95 **Sale $12.95**

IBM, Apple, Canon, Commodore, Atari are trademarks of International Business Machines, Apple Computer, Canon Inc, Commodore Business Machines, Atari Inc. Respectively.

Add $7.50 for shipping, handling and insurance. Illinois residents please add 6½% tax. Add $15.00 for CANADA, PUERTO RICO, HAWAII, ALASKA and APO-FPO orders. All orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Prices & Availability subject to change without notice.
VISA — MASTER CARD — C.O.D. C.O.D. on phone orders only

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

PROTECTO ENTERPRIZES

"The Computer Experts"

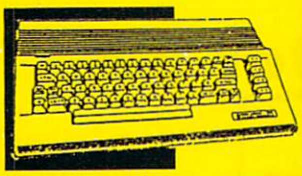
22292 N. Pepper Rd., Barrington IL 60010

• OUR WARRANTY •

All our products carry a minimum 90 day warranty from the date of purchase. If problems arise, simply send your product to us via U.P.S. prepaid. We will IMMEDIATELY send you a replacement at no charge via U.P.S. prepaid. This proves once again that...

We Love Our Customers!

COMMODORE 64c



Includes the GEOS program.

SALE $ 159.95

List $299

BLUE CHIP DISK DRIVE



SALE $159.95

List $249

COMMODORE 128



SALE $ 289.00

List $399

1571 DISK DRIVE



SALE $ 259.95

List $349

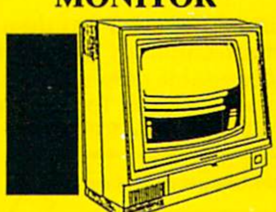
TV TUNER

Now switch your computer monitor into a television set with the flick of a switch. This Tuner has dual UHF/VHF selector switches, mute, automatic fine tuning and computer/TV selector switches. Hooks up between your computer and monitor! Inputs included for 300 ohm, 75 ohm, and UHF.

SALE $ 49.95

List $130

13" COLOR MONITOR



High Resolution, 1000 character display, with built in audio speaker with volume control.

SALE $ 179.95

List $329

BIG BLUE PRINTER



This is the affordable printer you've waited for! 8½" letter size, 80 column dot matrix, heat transfer printer features upper and lower case, underline, graphics, word processing, and much more.

SALE $ 39.95

List $199

PRINTER & TYPEWRITER COMBINATION

Superb Silver Reed letter quality daisy wheel printer/typewriter, just a flick of the switch to interchange. Extra large carriage, typewriter keyboard, automatic margin control, compact, lightweight, drop in cassette ribbon! Includes Centronics Parallel Interface

SALE $179.95

List $299

14" RGB & COMPOSITE COLOR MONITOR



High Resolution, 80 column Monitor. Switch from RGB to Composite. (C128 - IBM -Apple) RGB cable $19.95. Add $14.50 shipping.

SALE $ 237.00

List $399

12" 80 COLUMN MONITOR



High Resolution amber or green screen monitor. 80 col. x 1000 lines at center. Non-glare screen. Works terrific with IBM, Apple and Laser business computers.

SALE $ 99.00

List $129

COMSTAR 1000 PRINTER



Print letters, documents, etc., at 100 cps. Works in Near Letter Quality mode. Features are dot addressable graphics, adjustable tractor and friction feed, margin settings, pica, elite, condensed, italics, super/subscript, underline, & more. CBM Interface Included.

SALE $179.95

List $349

160-180 CPS N.L.Q. 180 PRINTER

This printer has a Near Letter Quality button on the front panel. No more turning the printer on and off. The 8K buffer will free up your computer four times faster than conventional printers and the high speed will keep you computing more than printing. Super graphics along with Pica, Elite, Italics, and Condensed print. **Lifetime Warranty** on Print Head plus 6 month immediate replacement warranty.

SALE $ 199.00

List $499

MasterCard

TO ORDER CALL (312) 382-5244

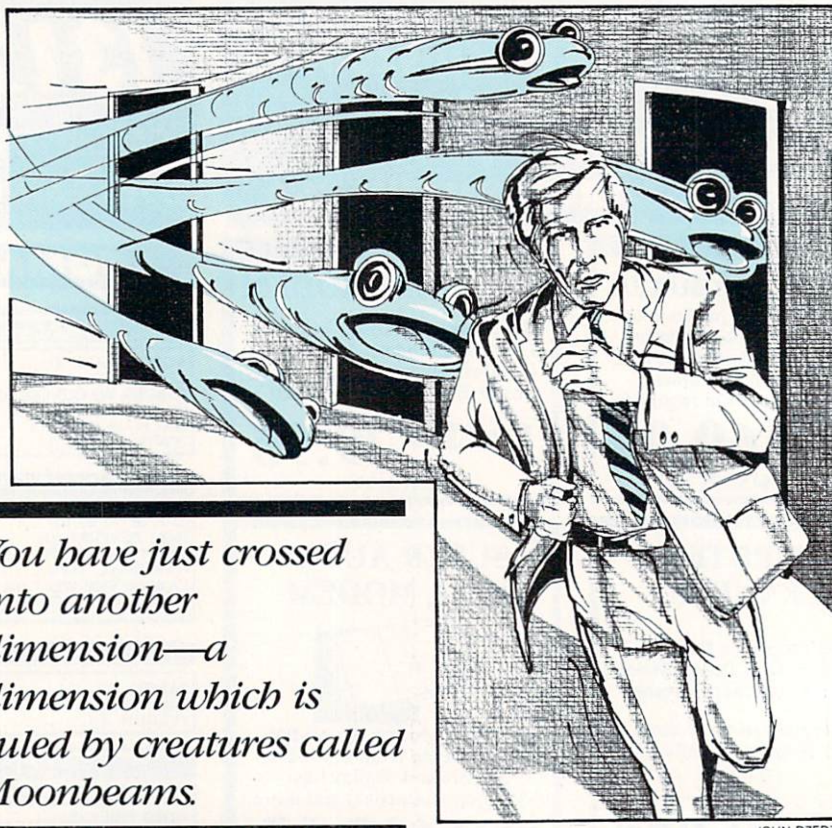
8 am - 8 pm CST Weekdays / 9 am - 12 noon CST Saturdays

VISA

BY JAMES C. HILTY

The Moonlight Zone

for the Commodore 64



JOHN DZEDZY

You have just crossed into another dimension—a dimension which is ruled by creatures called Moonbeams.

You are an average citizen living in an apartment located on the fiftieth floor of a new apartment building. It is a typical evening. Everything appears quiet and you are just about ready to settle back to read the newspaper and watch television.

Suddenly the lights go off and the television set goes blank. You assume that it is just a temporary power failure. But then the apartment highrise begins to sway. Spinning creatures appear out of nowhere. The creatures start to laugh and proclaim that you will never be able to leave the apartment building. Are you dreaming? Could these creatures be just part of your imagination?

The answer is no. You have just crossed over into another dimension—a dimension which is ruled by creatures called Moonbeams. You see, you have just crossed over into the Moonlight Zone!

You are not a person who gives up easily. You are determined to make it to the first floor of the building and escape. You must pass through the openings on each floor down to the next. The situation, however, is worse than you thought. The Moonbeams are roaming the building. If a Moonbeam catches you, he carries you up five floors and you are blopped. In addition, the girders of the apartment complex have become electrified—if you touch one you are also blopped! You must not give up. You must get out of the building.

This is the setting for The Moonlight Zone for the Commodore 64. After the title screen and familiar theme music, the playing screen appears. You are the man standing on the fiftieth floor. The Moonbeams are spinning everywhere. Using the joystick you can move your man left, right or down. Once you start moving left or right, you continue in that direction until you change direction with the joystick.

To move down to the next floor you must push the joystick down and move through one of the openings without hitting a girder. If you hit a girder or Moonbeam you are blopped. If you are blopped five times, the game is over. If the time runs out before you reach the first floor, the game is also over.

When the game is completed, your score is calculated based on whether or not you reached the first floor, how many times you were blopped, and the time remaining. The floor you are on, how many times you were blopped, and the high score are constantly displayed on the screen.

When you push the joystick down, your player does not move down—part of the screen moves up and a new floor is drawn on the bottom of the screen. This gives the appearance of the player moving down.

The Moonbeams roam in various directions. When combined with the illusion of the player moving down, the movement of the Moonbeams appears faster than what it really is. At times, the Moonbeams appear to move in a direction faster than they really are and may end up somewhere you don't expect them to be. Overcoming this optical illusion means a good bit in the strategy of the game.

No two apartment buildings are ever the same in The Moonlight Zone. Each floor is 39 characters long. These 39 characters are taken from a string of 255 characters with a random starting point for each floor. At any given time, four floors are displayed on the screen. Sometimes the openings in the floors may line up so that you can continue moving down without delay. At times you may wish to remain on a floor until the Moonbeams pass.

Remember, you must make it to the first floor before time runs out to escape. If a Moonbeam grabs you, you are transported up five floors and blopped. It is possible to end a game on a floor higher than the fiftieth. Unfortunately, this also means a very low score.

The Moonbeams and the player are sprites. The Moonbeams rotate through six sprite shapes (pointers) and the player rotates through two. The animation enhances the movement and makes for a more colorful and appealing game.

The Moonbeams are very pleased that you have arrived in the Moonlight Zone and wish to see you remain there. Use your skill and try to escape this strange dimension where Moonbeams roam and life will never be quite the same! ◼

Continued on pg. 50

Continued from pg. 48

Before typing these programs, read "How to Enter Programs," and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

The Moonlight Zone

```
14 S=54272:V=53248:TM=2500:SC=0:BL=0
   :FL=50:HS=0:P=200:Q=202'JWHS
20 GOSUB 800'BDLY
22 P$="":A$="[SHFT V4,SPACE4,SHFT V4,
   SPACE4,SHFT V14,SPACE4,SHFT V2,
   SPACE6,SHFT V10,SPACE4,SHFT V4]
   "'CFHM
23 B$="[SPACE5,SHFT V19,SPACE4,
   SHFT V4,SPACE4,SHFT V8,SPACE4,
   SHFT V6,SPACE4,SHFT V2,SPACE4]
   "'BCXM
24 C$="[SHFT V4,SPACE4,SHFT V4,SPACE4,
   SHFT V4,SPACE4,SHFT V4,SPACE4,
   SHFT V22,SPACE8,SHFT V3]"'BCKP
25 D$="[SHFT V7,SPACE4,SHFT V18,
   SPACE4,SHFT V6,SPACE4,SHFT V12,
   SPACE4,SHFT V4,SPACE3]"'BCRU
26 FOR X=50880 TO 51116:READ A
   :POKE X,A:NEXT'GSIL
28 FOR X=49152 TO 49528:READ A
   :POKE X,A:NEXT'GSWN
30 FOR X=12800 TO 13374:READ A
   :POKE X,A:NEXT'GSCG
32 P$=A$+B$+C$+D$'EKOF
34 POKE V+28,128:POKE V+37,1
   :POKE V+38,8:POKE V+46,14
   :POKE V+39,2:POKE V+40,7'MNES
35 POKE V+41,14:POKE V+42,5
   :POKE V+43,8:POKE V+44,15
   :POKE V+45,1'KGJR
36 FOR X=2040 TO 2046:POKE X,202:NEXT
   :POKE 2047,200:POKE V+16,84'IHUQ
37 POKE V+0,30:POKE V+1,20
   :POKE V+2,160:POKE V+3,20
   :POKE V+4,40:POKE V+5,20'MLCV
38 POKE V+10,30:POKE V+11,250
   :POKE V+6,100:POKE V+7,250
   :POKE V+8,10:POKE V+9,250'MQKX
39 POKE V+12,40:POKE V+13,250
   :POKE V+14,160:POKE V+15,130
   :POKE V+27,127'KNNW
40 S=50433:POKE S,3:POKE S+1,2
   :POKE S+2,0:POKE S+3,3:POKE S+4,252
   :POKE S+5,1'MMXP
41 POKE S+6,0:POKE S+7,255:POKE S+8,0
   :POKE S+9,254:POKE S+10,2'KEAN
42 POKE S+11,254:POKE S+12,254
   :POKE S+13,255:POKE S+14,0
   :POKE S+15,0'KKFP
44 W1=PEEK(V+30):W2=PEEK(V+31)'GPFL
50 REM    PLAYING SCREEN'BNXF
52 PRINT"[CLEAR,RVS,GREEN,SPACE3]
   T H E[SPACE2]M O O N L I G H T
   [SPACE2]Z O N E[SPACE3]"'BABM
54 PRINT"[RVS,YELLOW] TIME[SPACE26]
   BLOPS[SPACE3]"'BASM
56 PRINT"[YELLOW,SHFT V,RVS,GREEN]
   FLOOR[SPACE14]HIGH SCORE[SPACE8,
   RVOFF,YELLOW,SHFT V]"'BAIS
58 PRINT"[GREEN,SHFT V]";MID$(P$,
   INT(215*RND(9)+1),37);"[SHFT V]
   "'GTYS
60 FOR X=0 TO 3:FOR Y=0 TO 3
   :PRINT"[SHFT V]"TAB(38)"[SHFT V]"
   :NEXT Y'JNSM
61 PRINT"[SHFT V]";MID$(P$,
   INT(215*RND(9)+1),37);"[SHFT V]"
   :NEXT X'HVON
62 PRINT"[HOME,DOWN,RVS,YELLOW]
   "TAB(6);TM;TAB(36);BL
   :PRINT"[RVS,GREEN]"TAB(7);FL;
   TAB(31);HS'GACO
63 POKE V+21,255:FOR T=0 TO 1000
   :NEXT'GPLL
64 POKE 50688,255:POKE 50432,255
   :SYS 51104'DALM
65 JY=PEEK(56321)AND 15'DLAL
66 IF JY=7 THEN POKE 50447,1
   :POKE 50688,128'FUNP
67 IF JY=11 THEN POKE 50447,255
   :POKE 50688,128'FXNQ
68 IF JY=13 THEN 150'DHDN
70 P=P+1:IF P=202 THEN P=200'GMTJ
71 Q=Q+1:IF Q=208 THEN Q=202'GMGK
72 POKE 2047,P:FOR X=2040 TO 2046
   :POKE X,Q:NEXT'GVKN
73 W1=PEEK(V+30):IF W1>127 THEN
   200'GQDN
74 W2=PEEK(V+31):IF W2>127 THEN
   200'GQGO
75 TM=TM-5:PRINT"[HOME,DOWN,YELLOW,
   RVS]"TAB(6)"[SPACE5]"
   :PRINT"[HOME,DOWN,RVS]"TAB(6);TM
   :IF TM=0 THEN 300'JVJU
76 GOTO 65'BCSJ
149 REM    GOING DOWN'BJIL
150 POKE 50447,0:POKE 50688,128
    :POKE 49522,3:U=0'EDOI
152 SYS 49152:U=U+1:IF U=5 THEN
    160'GPHJ
153 W1=PEEK(V+30):IF W1 AND 128 THEN
    200'GQCK
154 W2=PEEK(V+31):IF W2 AND 128 THEN
    200'GQFL
156 GOTO 152'BDHH
160 PRINT"[HOME,GREEN,DOWN22]"'BAND
161 PRINT"[SHFT V]";MID$(P$,
    INT(215*RND(9)+1),37);"[SHFT V]
    "'GTTL
162 FL=FL-1:PRINT"[HOME,DOWN2,RVS,
    GREEN]"TAB(7)"[SPACE3]"
    :PRINT"[HOME,DOWN2,RVS]"TAB(7);FL
    :IF FL=1 THEN 320'JVWO
165 GOTO 70'BCOG
199 REM    COLLISIONS'BKQQ
200 POKE 50432,0:FOR X=2040 TO 2046
    :POKE X,208:NEXT'GYAE
202 FOR X=0 TO 24:POKE S+X,0:NEXT'GKDE
204 POKE S+24,15:POKE S+5,80
```

```
    :POKE S+6,243:POKE S+3,4
    :POKE S+4,65'KFEM
206 FOR X=20 TO 140 STEP 5:POKE S+1,X
    :NEXT'HNQJ
207 POKE S+4,64:FOR X=0 TO 50:NEXT
    :POKE S+4,64'IRAL
208 FOR X=0 TO 24:POKE S+X,0:NEXT'GKDK
210 POKE S+24,143:POKE S+5,16
    :POKE S+19,16:POKE S+6,252
    :POKE S+20,249'KKBK
212 POKE S+4,21:POKE S+18,17
    :POKE S+1,68:POKE S+15,42
    :FOR X=0 TO 200:NEXT'MHDN
214 POKE S+24,20:POKE S+18,16
    :FOR X=0 TO 400:NEXT:FOR X=0 TO 24
    :POKE S+X,0:NEXT'OGTQ
218 FL=FL+5:BL=BL+1:IF BL=5 THEN
    370'HSTO
220 PRINT"[CLEAR]":POKE V+21,0
    :GOTO 36'EJHC
299 REM      TIME RUNS OUT'BLDR
300 POKE 50432,127:PRINT"[HOME,DOWN9]
    "TAB(10)"SORRY..TIME RAN
    OUT."'DNMH
302 GOSUB 500:SC=1000-(50*BL)-(FL*10)
    :GOTO 400'HBQI
319 REM      FIRST FLOOR'BKCK
320 POKE 50432,127:PRINT"[HOME,DOWN5,
    CYAN]":PRINT TAB(5)
    "CONGRATULATIONS! YOU
    ESCAPED"'ENRM
322 GOSUB 500:SC=1000+(2*TM)-(50*BL)
    :GOTO 400'HAIK
369 REM      BLOPPED'BHXO
370 POKE 50432,127:PRINT"[HOME,CYAN,
    DOWN13]"'CKOI
372 PRINT TAB(2)"SORRY..YOU WERE
    BLOPPED FIVE TIMES!"'CCAR
374 GOSUB 500:SC=750-(FL*10)
    :IF SC<0 THEN SC=0:GOTO 400'JBRT
399 REM      GAME OVER'BIDR
400 PRINT"[CLEAR]":POKE V+21,127
    :FOR X=2040 TO 2046:POKE X,202
    :NEXT:Q=202'JFBK
402 PRINT"[DOWN8,CYAN]"TAB(11)"[RVS]
    G A M E[SPACE2]O V E R":PRINT
    :PRINT:PRINT:PRINT'GHUJ
404 PRINT TAB(14)"SCORE"SC:PRINT
    :IF SC>HS THEN HS=SC'HPDL
405 PRINT TAB(14)"HIGH SCORE"HS'CFQI
406 PRINT"[DOWN6]"TAB(11)"PLAY AGAIN
    [SPACE2]Y OR N"'CDIK
408 GET JUNK$:IF JUNK$<>""THEN
    408'FOVM
410 GET M$:IF M$=""THEN 416'EIJD
412 IF M$="Y"THEN 420'DFSE
414 END'BACD
416 Q=Q+1:IF Q=208 THEN Q=202'GMGL
```

Continued on pg. 52

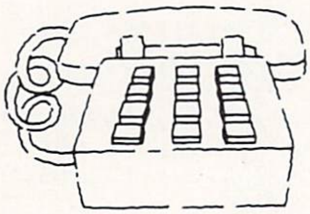
```
417 FOR X=2040 TO 2046:POKE X,Q:NEXT
    :GOTO 410'GSWN
420 SC=0:TM=2500:FL=50:BL=0
    :PRINT"[CLEAR]":POKE V+21,0
    :POKE 50432,0:GOTO 36'JMQO
499 REM    MUSIC'BFQS
500 RESTORE'BAOY
502 FOR L=S TO S+24:POKE L,0:NEXT
    :POKE S+5,9:POKE S+6,0
    :POKE S+24,15'MCFO
504 READ HF,LF,DR:IF HF=0 THEN
    RETURN'FMEI
506 POKE S+1,HF:POKE S,LF:POKE S+4,33
    :FOR T=1 TO DR:NEXT'JWUP
508 POKE S+4,32:FOR T=1 TO 50
    :NEXT'GLQN
510 GOTO 504'BDIB
512 DATA 44,193,250,63,75,100,56,99,
    100,59,190,100,53,57,100'BBNM
514 DATA 63,75,100,56,99,100,59,190,
    100,53,57,100'BPGM
516 DATA 63,75,100,56,99,100,59,190,
    100,53,57,100'BPGO
518 DATA 63,75,100,56,99,100,59,190,
    100,53,57,100'BPGQ
520 DATA 44,193,250,84,125,1000,0,0,
    0'BDLG
799 REM    TITLE'BFRV
800 PRINT"[CLEAR]":POKE 53281,0
    :POKE 53280,0'DQOH
802 FOR T=0 TO 100:X=INT(1000*RND(9)
    +1024):POKE X,46:NEXT'KBLP
804 PRINT"[DOWN4,YELLOW]
    "TAB(17)"T H E":PRINT
    :PRINT TAB(11)"M O O N L I G H T"
    :PRINT'GJCR
806 PRINT TAB(10)"[CMDR P5]"TAB(25)"
    [SHFT O,CMDR Y4]":PRINT TAB(14)"
    [SHFT N]"TAB(25)"[CMDR H]"'GNLW
808 PRINT TAB(13)"[SHFT N,SPACE2,
    SHFT O,CMDR Y,SHFT P]"TAB(25)"
    [SHFT O,CMDR Y2]"'DGNS
810 PRINT TAB(12)"[SHFT N,SPACE3,
    CMDR H] [CMDR N2,SHFT M,SPACE2,
    CMDR H] [CMDR H]"'CDKK
812 PRINT TAB(11)"[SHFT N,SPACE4,
    CMDR H] [CMDR N2] [SHFT M]
    [CMDR H] [SHFT L,CMDR P3]"'CDBP
814 PRINT TAB(10)"[SHFT N,SPACE5,
    SHFT L,CMDR P,SHFT @,CMDR N,
    SPACE2,SHFT M,CMDR H]"
    :PRINT TAB(9)"[SHFT N]"
    :PRINT TAB(9)"[CMDR Y6]"'GJTX
816 PRINT"[DOWN2]"TAB(15)"[CYAN]
    BY JC HILTY"'CDKO
818 GOSUB 500'BDIM
820 PRINT"[DOWN2]"TAB(7)"[YELLOW]
    PLEASE WAIT..READING DATA"
    :RETURN'DDSN
1000 REM    SPRITE MOTION'BMJW
1002 DATA 169,255,45,0,198,240,16,169,
     0,141,0,198,162,21,189,0'BCTF
1003 DATA 197,157,0,198,202,208,247,
     162,1,169,1,141,80,197,173,80,
     197'BJBH
1004 DATA 45,0,197,240,3,76,243,198,
     232,232,14,80,197,208,238,76,49,
     234'BLUJ
1006 DATA 169,0,29,0,197,208,3,76,97,
     199,169,128,61'BQQH
1007 DATA 0,197,240,48,254,0,198,208,
     40,222,255,207,76,144,199,80'BFXL
1008 DATA 197,45,16,208,208,12,173,16,
     208,13,80,197,141,16,208,76'BFSM
1009 DATA 43,199,173,16,208,77,80,197,
     141,16,208,189,0,197,157,0'BELN
1010 DATA 198,76,97,199,222,0,198,208,
     40,254,255,207,208,29,173,80'BGSF
1011 DATA 197,45,16,208,208,12,173,16,
     208,13,80,197,141,16,208,76'BFSG
1012 DATA 91,199,173,16,208,77,80,197,
     141,16,208,189,0,197,157,0'BEOH
1013 DATA 198,169,0,232,29,0,197,208,
     3,76,140,199,169,128,61,0'BCJH
1014 DATA 197,240,11,254,0,198,208,20,
     222,255,207,76,134,199,222,0'BGGJ
1015 DATA 198,208,9,254,255,207,189,0,
     197,157,0,198,202,76,233,198'BGUK
1016 DATA 169,255,221,255,207,240,3,
     76,43,199,173,80,197,76,17,
     199'BGUL
1017 DATA 120,169,192,141,20,3,169,
     198,141,21,3,88,96'BSRJ
1018 REM    SCREEN MOVEMENT'BOKG
1019 DATA 174,114,193,224,3,144,3,76,
     117,192,188,114,193,140,121,193,
     174'BMDP
1020 DATA 118,193,232,202,32,30,193,
     172,121,193,173,119,193,201,2,
     208,10'BMDH
1021 DATA 169,32,72,173,33,208,72,76,
     50,192,177,90,72,177,92,72,204,
     116'BLVI
1022 DATA 193,240,20,200,177,90,72,
     177,92,136,145,92,104,145,90,200,
     204'BLRJ
1023 DATA 116,193,208,238,240,18,136,
     177,90,72,177,92,200,145,92,104,
     145'BMLK
1024 DATA 90,136,204,115,193,208,238,
     173,119,193,201,0,208,5,104,104,
     76'BLQL
1025 DATA 111,192,104,145,92,104,145,
     90,236,117,193,208,160,96,172,
     116,193'BOCM
1026 DATA 200,189,114,193,170,32,30,
     193,173,120,193,201,2,208,19,136,
     169'BMQN
1027 DATA 32,153,122,193,173,33,208,
     153,162,193,204,115,193,208,239,
     240'BLWO
1028 DATA 16,136,177,90,153,122,193,
     177,92,153,162,193,204,115,193,
     208,240'BOIP
```

```
1029 DATA 236,117,193,240,37,202,32,
     30,193,172,116,193,200,136,177,
     90,72'BMUQ
1030 DATA 177,92,32,48,193,145,92,104,
     145,90,32,56,193,204,115,193,
     208'BKPI
1031 DATA 234,236,117,193,208,221,240,
     46,202,206,118,193,232,32,30,193,
     172'BOMJ
1032 DATA 116,193,200,136,32,48,193,
     177,90,72,177,92,32,56,193,145,
     92,104'BNQK
1033 DATA 145,90,204,115,193,208,234,
     236,118,193,208,221,238,118,193,
     232'BMFL
1034 DATA 32,30,193,173,120,193,201,0,
     240,20,172,115,193,136,200,185,
     162'BMTM
1035 DATA 193,145,92,185,122,193,145,
     90,204,116,193,208,240,96,189,89,
     193'BNJN
1036 DATA 133,91,24,105,212,133,93,
     189,64,193,133,90,133,92,96,72,
     152'BJRN
1037 DATA 24,105,40,168,104,96,72,152,
     56,233,40,168,104,96,0,40,80'BGBO
1038 DATA 120,160,200,240,24,64,104,
     144,184,224,8,48,88,128,168,208,
     248'BLCQ
1039 DATA 32,72,112,152,192,4,4,4,4,4,
     4,4,5,5,5,5,5'BQCN
1040 DATA 5,6,6,6,6,6,6,6,7,7,7,7,7,3,
     1,37,3,23,2,2'BQHF
1042 REM    SPRITE DATA'BKCC
1044 DATA 0,0,0,0,60,0,0,52,0,0,20,0,
     0,16,0,0,40,0,0,168,0,0,170,64,0,
     168'BNGN
1045 DATA 0,0,156,0,0,40,0,0,40,0,0,
     130,0,0,130,0,0,243,192,0,0,0,0,
     0,0'BLUN
1046 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0'BIOJ
1047 DATA 60,0,0,52,0,0,20,0,0,16,0,0,
     40,0,0,168,0,0,168,0,0,168,0,0,
     188'BMBQ
1048 DATA 0,0,104,0,0,40,0,0,40,0,0,
     40,0,0,63,0,0,0,0,0,0,0,0,0,0,
     0,0'BLKQ
1049 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
     0'BCTL
1050 DATA 255,0,1,255,128,3,255,192,7,
     255,224,14,60,112,30,60,120,62,
     60'BLUK
1051 DATA 124,127,255,254,255,255,255,
     255,255,255,248,255,31,120,255,
     30'BLLL
1052 DATA 56,0,28,28,0,56,14,0,112,7,
     255,224,3,255,192,1,255,128,0,
     255,0,0'BOMM
1053 DATA 0,0,0,0,0,0,0,255,0,1,255,
     128,3,255,192,7,255,224,15,241,
     192,31'BNRN
1054 DATA 241,192,63,241,196,127,255,
     254,255,255,255,255,248,255,255,
     248,0'BOAO
1055 DATA 127,252,0,63,254,0,31,255,0,
     15,255,128,7,255,224,3,255'BEYN
1056 DATA 192,1,255,128,0,255,0,0,0,0,
     0,0,0,0,0'BMGL
1057 DATA 255,0,1,255,128,3,255,192,7,
     255,224,15,255,192,31,255,192,
     63'BKCR
1058 DATA 255,196,127,255,254,255,255,
     255,255,255,31,255,255,0,127,
     255'BKYS
1059 DATA 128,63,255,192,31,255,224,
     15,255,240,7,255,224,3,255,
     192'BGNS
1060 DATA 1,255,128,0,255,0,0'BTFD
1061 DATA 0,0,0,0,0,0,0,255,0,1,255,
     128,3,255,192,7,255,224,15,255,
     240'BKDL
1062 DATA 31,255,248,63,255,252,127,
     255,254,255,255,255,255,255,
     255'BHJM
1063 DATA 255,255,255,127,255,254,63,
     255,252,31,255,248,15,255,240,
     7'BIXN
1064 DATA 255,224,3,255,192,1,255,128,
     0,255,0,0,0,0,0,0,0,0,255,0,
     1'BJIO
1065 DATA 255,128,3,255,192,7,255,224,
     3,255,240,3,255,248,35,255,252,
     127'BMHQ
1066 DATA 255,254,255,255,255,248,255,
     255,0,255,255,1,255,254,3,255,
     252'BLQR
1067 DATA 7,255,248,15,255,240,7,255,
     224,3,255,192,1,255,128,0,255,0,
     0,0,0'BOWS
1068 DATA 0,0,0,0,0,255,0,1,255,128,3,
     255,192,7,255,224,7,199,240,7,
     199'BLET
1069 DATA 248,7,199,252,127,255,254,
     255,255,255,255,199,255,0,7,255,
     0,7'BLFU
1070 DATA 254,0,15,252,0,31,248,15,
     255,240,7,255,224,3,255,192,1,
     255,128'BMIM
1071 DATA 0,255,0,0,0,0,63,248,0,63,
     249,128,51,153,159,51,153,155,51,
     249'BMFN
1072 DATA 155,63,241,155,63,193,155,
     48,241,155,48,249,159,48,153,128,
     48'BLJO
1073 DATA 153,240,48,249,240,0,248,0,
     0,0,0,0,0,0,0,15,255,240,31,
     255'BIYO
1074 DATA 224,63,255,192,0,0,0,0,0,
     0'BBDJ
```

END

TEVEX

CHALLENGING SOFTWARE AT DISCOUNT PRICES WITH QUALITY SERVICE. WHAT AN UNBEATABLE COMBINATION!



CALL TOLL-FREE

For your convenience we have toll-free ordering. Whether you're calling to place an order or just to ask a question, every call is always welcome on our toll-free lines.



SAME DAY SHIPPING

We ship every order the same day it's placed. Just call before 3:30 and we'll ship your order via UPS. US Mail Service also available.



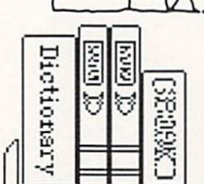
DISCOUNT PRICES

Save up to 1/3 off the retail price when you buy from Tevex. Computer software is expensive. Why pay more than you have to for the same product.



FRIENDLY, KNOWLEDGEABLE STAFF

At Tevex, our staff is always available to answer your questions. We keep up with all the new releases and upcoming games. Tevex is the information link you need to stay informed about the latest in computer software.



For Your Information

At Tevex, we have a full line of helpful publications for your use. Twice each year we send out our software catalog. These are 40 pages or more, packed with information. Each catalog has more than 250 software reviews. For each product covered, you can read about the complexity of the game, the number of players, subject matter, graphics, special features, playing time, and more.

The best part is that these catalogs are FREE with your first order from TEVEX. We publish the catalogs in the spring and in the fall. You can also get one just by sending us your name and address and we'll put you on our mailing list.

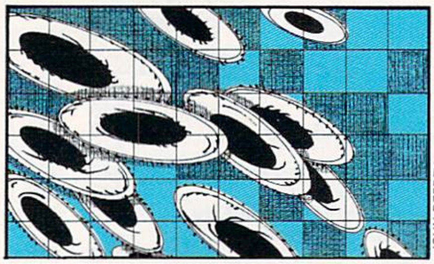
WRITE FOR FREE CATALOG
Open 9-8 Mon-Fri.
10-5 Sat.

BY MARTY PLUMBO, JR.

AmigaLife

for the Amiga

The Rocky Horror Picture Show, the Grateful Dead, Kurt Vonnegut—practically every form of artistic expression has its share of cult favorites, and computer gaming is no exception. Back in 1971, Martin Gardner of *Scientific American* magazine introduced to an unsuspecting world a computer simulation of growth and development in populations that had been created a year earlier by University of Cambridge mathematician John H. Conway. The game was Life, and in the decade and a half since its invention, Life has become something of a computer phenomenon.


JOHN DZEDZY

Play the game of Life on your Amiga.

About Life

Life is more than an experiment requiring observation; it requires participation. In the simulation, a group of "cells" are arranged on a grid, and through the course of the game, an infinite number of generations will live, die and reproduce according to a simple set of rules.

When Conway first set out to devise the rules for Life, he did so with respect to what he felt to be three very basic genetic principles regarding populations, which he outlined as follows.

1. There should be no initial pattern for which there exists a simple proof that the population could undergo infinite growth.
2. There should be initial patterns that seem to undergo infinite growth.
3. There should be initial patterns that grow and develop for many generations, ultimately concluding in one of three conditions: complete barrenness; a single, unchanging arrangement of cells; or oscillation between two or more different arrangements of cells.

This particular set of rules is brilliant, not only because of its simplicity, but because of the infinitely unpredictable, even seemingly random, behavior it creates. To implement these rules in a program, Conway created the Life universe, a finite grid in which each cell represents either a dead or living organism. Each cell has eight neighboring cells, four diagonally adjacent and four orthogonally adjacent. The rules of growth and development for each cell depend exclusively on the number of neighboring cells which are living.

1. Birth—If an empty cell is adjacent to exactly three living cells, a new cell is born there.
2. Survival—If a living cell is adjacent to either two or three living cells, it survives to the next generation.
3. Death—If a living cell is adjacent to four or more cells, it perishes from overcrowding. If a living cell is adjacent to one or no cells, it succumbs to isolation.

As a population grows, it develops according to these rules, unusual and unexpected variations.

Playing Life

After typing in AmigaLife, save it to disk one or two times, just as a precaution, before you run it. When you first run AmigaLife, the screen will go black for about 25 seconds while the program initializes. When it's done, the Life grid will be displayed, and you will be able to begin setting up your initial pattern. Just move the mouse pointer to any of the 420 cells on the grid, pressing the left mouse button to ordain it with life. If you should change your mind about a particular cell, just point and press over it again, and it will no longer be alive.

Once you're satisfied with the arrangement, just point to the BEGIN box and press the left mouse button, then sit back and watch the show. All births and deaths will occur simultaneously with each passing generation as the total population and present generation are displayed in the upper left-hand corner.

Remembering Conway's third genetic assumption, you will note that only rarely does a population completely die out. Most initial patterns reach either stable patterns, consisting of a single, unchanging arrangement of cells—such as a hexagon formed from six living cells or patterns that oscillate forever between two or more different arrangements or living cells.

As you use AmigaLife, you will discover that the challenge lies in determining which patterns will grow in number and complexity with successive generations, and which patterns will simple die out. The computer will update the pattern with each passing generation, every couple of seconds, continuing for as few as a handful of distinct generations, as many as an infinite number of generations, or until the entire population dies off completely. You can experiment.

It should be noted that AmigaLife does not allow patterns to grow beyond the boundaries of the grid.

Something Extra

Life is traditionally an exercise in spectatorship since the player is only required to interact with the program during the initial setup phase. After that, the computer does everything. In AmigaLife I have added an option that I have long missed in the classic version of Life.

If at any time during the growth phase of a session of AmigaLife you should decide to alter any portion of the grid, just point the mouse cursor at the SETUP box and press the left mouse button. The program will re-enter the setup phase, only without erasing the existing pattern. Now you can determine whether each cell will be living or nonliving, just as before. Once you're done changing the pattern, just select the BEGIN box to resume play.

One of the first things that you'll notice while experimenting with AmigaLife is that the patterns produced by the program have a tendency towards symmetry. Populations initially nonsymmetrical tend to evolve irreversibly into symmetrical ones, while populations that are initially symmetrical stay that way. Using the setup option, you can experiment with altering symmetry to produce some very interesting results. ■

AmigaLife

```
' AmigaLife V1.4
' AMIGA rendition of Conway's "Life"
' by Marty Plumbo
DEFINT a-z
```

```
DIM field1 (31,15), field2 (31,15) , p%(8)

FOR x=0 TO 8 ' define SAY parameters
READ p%(x)
NEXT
DATA 110,0,150,0,22200,64,10,1,0

SAY "WAH3N MOW4MEHNT PLIY4Z.",p%

FOR x=0 TO 3 ' darken screen
PALETTE x,0,0,0
NEXT

WINDOW 2, "AmigaLife",(0,0)-(631,186),30
GOSUB Drawfield
WINDOW 1, "AmigaLife",(0,0)-(631,186),30
GOSUB Drawfield

PALETTE 0,0,0,1 ' re-color the screens
PALETTE 1,1,1,1
PALETTE 2,0,0,0
PALETTE 3,0,1,0

SAY "THAE4NXK YUW.",p%

Blah:
  SAY "YUW MEY NAW AXRREY3NJ DHAX BOW4RD.",p%

Setup:
  WHILE MOUSE(0)=0 ' wait for mouse button
  WEND

  m1=MOUSE(1)
  m2=MOUSE(2)
  IF m2>11 AND m2<27 AND m1>514 AND m1<597 THEN Play
  IF m1<16 OR m1>611 OR m2<39 OR m2>177 THEN Setup
  x=INT((m1-16)/20)+1 ' flip cell in field2
  y=INT((m2-39)/10)+1
  field2(x,y)=1-field2(x,y)
  GOSUB Blink
  GOSUB Mwait
  GOTO Setup

Play:
  FOR x=1 TO 2
  AREA (515,12)
  AREA (596,12)
  AREA (596,26)
  AREA (512,26)
  AREAFILL 1
  FOR y=1 TO 500
  NEXT
  NEXT

  BEEP
  ON MOUSE GOSUB Setgo
  GOSUB Mwait

Loop:
  FOR w=1 TO 2
  IF ww=2 THEN w=2
  WINDOW OUTPUT 3-w
  MOUSE ON ' interrupts are okay during this part
  ww=0
  p=0
  g=g+1
  GOSUB Drawgrid
  GOSUB Check
  FOR x=minx TO maxx
  FOR y=miny TO maxy
  a=field2(x,y)
  c=field2(x-1,y-1)+field2(x,y-1)+
     field2(x+1,y-1)+field2(x+1,y)+
     field2(x+1,y+1)+field2(x,y+1)+
     field2(x-1,y+1)+field2(x-1,y)
  field1(x,y)=0
  IF c=3 THEN field1(x,y)=1 : p=p+1 : GOSUB Blink
  NEXT
  NEXT

  COLOR 1 ' print generation and population totals
  LOCATE 3,19
  PRINT USING "###" ; g
  LOCATE 3,40
  PRINT USING "###" ; p
  MOUSE STOP ' no interrupts during this

  FOR x=0 TO 31
  FOR y=0 TO 15
  field2(x,y)=field1(x,y)
  NEXT
  NEXT

  WINDOW 3-w
  NEXT
  GOTO Loop

*********************************************
*                subroutines                *
*********************************************
Drawfield:
  COLOR 2
  LINE (420,12) - (501,26),,BF
  LINE (515,12) - (596,26),,BF
  LINE (16,12) - (362,26),,BF
  COLOR 1,2
  LOCATE 3,56
  PRINT "SETUP"
  LOCATE 3,68
  PRINT "BEGIN"
  LOCATE 3,5
  PRINT "Generation: 0 Population     0"
  LINE (424,14) - (497,24),,B
  LINE (519,14) - (592,24),,B
  LINE (20,14) - (358,24),,B

Blink:
  AREA (((x-1)*20)+14,((y-1)*10)+37)
  AREA STEP (21,0)
  AREA STEP (0,11)
  AREA STEP (-21,0)
  AREAFILL 1
  RETURN

Drawgrid:
  COLOR 3
  LINE (14,37) - (614,177),,BF
  COLOR 2
  FOR x=14 TO 614 STEP 20
  LINE (x-1,37) - (x+2,177),,BF
  NEXT
  FOR y=37 TO 177 STEP 10
  LINE (14,y) - (614,y)
  LINE (14,y+1) - (614,y+1)
  NEXT
  RETURN
```

GAME PROG/AMIGALIFE

```
Mwait:
   WHILE MOUSE (0)<>0
   WEND
   RETURN

Setgo:
   i1 = MOUSE(0)
   i1 = MOUSE(1)
   i2 = MOUSE(2)
   IF i2>11 AND i2<27 AND i1>419 AND i1<502 THEN
    MOUSE OFF
    FOR i1 = 1 TO 2
    AREA (420,12)
    AREA (501,12)
    AREA (501,26)
    AREA (420,26)
    AREAFILL 1
    FOR i2 = 1 TO 500
    NEXT
    NEXT
    BEEP
    WINDOW OUTPUT w

    IF w = 2 THEN ww = 2

    FOR x = 0 TO 31 ' clear scratch field
    FOR y = 0 TO 15
    field1(x,y) = 0
    NEXT
    NEXT

    GOSUB Mwait
    RETURN Blah

   END IF
   RETURN

Check:
  minx = 30
  maxx = 0

  FOR y = 1 TO 14
  FOR x = 1 TO 30
  IF field2(x,y) = 1 AND y>1 THEN miny = y-1 : GOTO c1
  NEXT
  NEXT

c1:
  FOR y = miny + 1 TO 14
  FOR x = 1 TO 30
  IF field2(x,y) = 1 AND x<minx AND x>1 THEN minx = x-1
  NEXT
  NEXT

  FOR y = 14 TO miny STEP -1
  FOR x = 30 TO 1 STEP -1
  IF field2(x,y) = 1 AND y<14 THEN maxy = y + 1 : GOTO c2
  NEXT
  NEXT

c2:
  FOR y = maxy-1 TO miny STEP -1
  FOR x-30 TO 1 STEP -1
  IF field2(x,y) = 1 AND x = >maxx AND x<30 THEN maxx = x + 1
  NEXT
  NEXT
  RETURN
```

END

Tips & Tricks

Continued from pg. 14

```
     Y="+STR$(Y)+"[SPACE3]"
5170 GOTO 5030
5180 GRAPHIC 0:END
```

END

Free Envelopes: If you use the *The Print Shop* with the MPS-801/1525 printer, you can save money two ways when printing greeting cards. Take an envelope which fits the cards you print, carefully break its seams and trace it on cardboard. Cut out the template and, using the usually wasted first sheet of paper wasted by your printer, trace around the template. Cut out the envelope and fold it to size. Two swipes of a glue stick produce a perfect envelope for your cards, and more glue stick seals it up for mailing. No more hunting for hard-to-find funny-sized envelopes. And no more wasting paper.
Patricia J. Kercel
Henderson, Nevada

Cheap Paper: If you use the Okimate 10 or other similar printer, you may have had trouble finding an accessible source of inexpensive paper. My local K-Mart sells Sharp heat-sensitive paper (EA-1LS5) for about $7 for a pack of 100 sheets. It works fine in my Okimate, and, of course, I don't need to use my ribbon when I'm printing in black only. ■

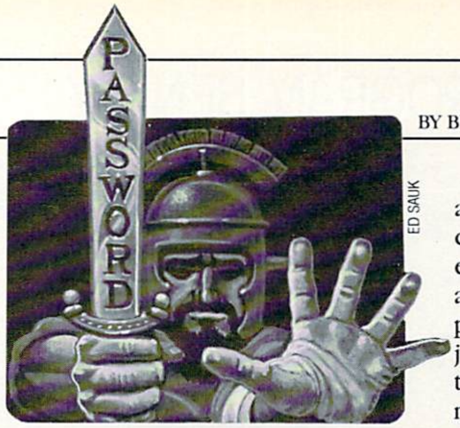
Joe Borysowicz
Cicero, Illinois

JIFFIES/TELEPHONE LISTER

Continued from pg. 30

```
630 OPEN 15,8,15'BHAE
640 PRINT#15,"S0:PHONE NUMBERS"'BDVJ
650 INPUT#15,E,E$ : IF E<>1 THEN
    PRINT E$ : STOP'HNCM
660 SAVE "0:PHONE NUMBERS",8'BCML
670 INPUT#15,E,E$ : IF E THEN PRINT
    E$ : STOP'FMIM
680 CLOSE 15 : END'CDQJ
690 REM'BARJ
700 REM NAME & NUMBER DATA'BPJF
710 REM PLACE A "/" BETWEEN NAME AND
    NUMBER,'BCMK
720 REM AND A "/" BETWEEN THE HOME
    AND'BVKJ
730 REM BUSINESS NUMBER'BOUI
740 REM'BARF
750 DATA "[SHFT F]URTER, [SHFT F]
    RANK/123-9876/234-5678"'BANO
760 DATA "[SHFT P]OCKETS, [SHFT M].
    [SHFT T]./(612)-123-4567/NONE"'BA
    TQ
770 DATA "[SHFT K]ID, [SHFT B]ILLY
    [SHFT T]./456-7890"'BASP
780 DATA "[SHFT D]'[SHFT P]RUNES,
    [SHFT S]TU/987-6543/333-3333"'BAVS
790 DATA "[SHFT S]TOVE, [SHFT F]
    RANKLIN/888-8888/123-4321"'BAQT
800 DATA "END"'BARD
```

END



BY BENNETT COOKSON, JR. AND ALAN W. POOLE

Program Sentry *for the Commodore 64*

Protect Your BASIC program with this password security system.

Protecting your program from unauthorized users isn't easy. Almost anyone can learn to load and list programs. Program Sentry will guard your BASIC programs with a password security system. Anyone who attempts to load a program without the correct password will only get random characters. It works with either tape or disk.

Since Program Sentry alters your program into a series of random numbers, breaking the code to recover the program is nearly impossible. Even an experienced machine-language programmer would find it very difficult. There are over 11 trillion possible passwords, and since the password is not stored, there is no way to discover the password from the program listing.

The BASIC program with this article places a machine-language program into memory starting at 52950. Make sure you type all the numbers correctly. One wrong number could ruin the program. Since the BASIC program erases itself when it is through, be sure to save it before you run it.

First, load and run Program Sentry. Next, load the program you want to protect. The computer will ask you to enter the password. Since this program has not been protected previously, there is no password yet. Just press RETURN without typing anything and the program will load as normal. After the program loads, type the usual SAVE command and press RETURN. You will probably want to give the program a new name to indicate that this copy is protected. The computer will then ask you what you want the password to be. Type a password and press RETURN. The password can be any length. However, only the first six characters are important. Any characters past the first six will be ignored.

The password can contain any type of character, including graphics characters. If you want to save the program without protecting it, just press RETURN without typing anything. If you really want your program to be safe, you will have to erase any copies of it that were saved without Program Sentry. Warning: Do not forget the password. There is no way to recover the program without the right password.

Load and run Program Sentry if is isn't already in memory. Type the normal load command. The computer will ask you to enter the password. Type the password and press RETURN. If you want to load a program that has not been protected, just press RETURN without typing anything. The program will then load into memory. If you typed the correct password, the program will be the same as it was originally. If you typed the wrong password, the program will not run and list only garbage on the screen.

If you want the computer to stop asking you for passwords, you can easily disconnect Program Sentry by holding down the RUN/STOP key and pressing RESTORE. However, without it you will not be able to use programs that have been protected with Program Sentry. Some programs may destroy the machine-language code by changing bytes from addresses 52950 through 53211. If this happens, or if you pressed RESTORE, you will have to load and run Program Sentry again to put it back in memory.

Program Sentry uses a random number generator that creates numbers from a six byte seed value. Given the same seed value, the series of random numbers will always be the same. The codes for the first six characters of the password are placed in the six bytes of the seed value. Before a program is saved, each byte is altered by performing an exclusive-OR operation on it with a random number, thus turning your program into random characters. When you load a program with the right password, each byte is again exclusive-ORed with the same series of random numbers, returning your program to its original form. ■

Before typing these programs, read "How to Enter Programs," and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Program Sentry

```
10 POKE 53280,0:POKE 53281,0'CPLB
20 PRINT CHR$(147):PRINT
   CHR$(154)'ELJC
30 FOR L=1 TO 9:PRINT:NEXT'FFCD
40 PRINT CHR$(18);'CFJC
50 PRINT TAB(13);"PROGRAM SENTRY"
   :PRINT:PRINT'EGLI
55 PRINT"BY BENNETT COOKSON JR. &
   ALAN W. POOLE"'BAKQ
60 A=52950'BGWD
70 READ N:IF N=256 THEN 110'EJLH
80 POKE A,N'BDGF
90 A=A+1:S=S+N'EHCJ
100 GOTO 70'BCOU
110 IF S<>27668 THEN PRINT
    :PRINT"THERE IS AN ERROR IN THE
    DATA.":STOP'HIPJ
120 POKE 818,214:POKE 819,206'CPYB
130 POKE 816,40:POKE 817,207'COBB
140 PRINT CHR$(19):PRINT
    CHR$(147)'EKLD
150 NEW'BALA
52950 DATA 72,138,72,152,72,169,11,
      160'BCPK
52958 DATA 207,32,30,171,32,196,207,
      32'BCES
52966 DATA 96,165,173,0,2,240,6,
      32'BXOQ
52974 DATA 207,207,32,126,207,104,168,
```

```
      104'BFWR
52982 DATA 170,104,32,237,245,144,1,
      96'BCJP
52990 DATA 173,0,2,240,6,32,207,
      207'BYXO
52998 DATA 32,126,207,24,96,13,87,
      72'BATW
53006 DATA 65,84,32,87,73,76,76,
      32'BXLD
53014 DATA 66,69,32,84,72,69,32,
      80'BXIC
53022 DATA 65,83,83,87,79,82,68,
      63'BXYB
53030 DATA 32,0,72,138,72,152,72,
      169'BAPA
53038 DATA 98,160,207,32,30,171,32,
      196'BCOI
53046 DATA 207,32,96,165,173,0,2,
      208'BALH
53054 DATA 8,104,168,104,170,104,76,
      165'BDMG
53062 DATA 244,32,207,207,104,168,104,
      170'BFWG
53070 DATA 104,32,165,244,144,1,96,
      134'BCIE
53078 DATA 45,132,46,32,126,207,166,
      45'BCNM
53086 DATA 164,46,24,96,13,80,76,
      69'BYGL
53094 DATA 65,83,69,32,69,78,84,
      69'BXAK
53102 DATA 82,32,84,72,69,32,80,
      65'BXCA
53110 DATA 83,83,87,79,82,68,58,0'BWYE
53118 DATA 165,43,133,253,165,44,133,
      254'BEHI
53126 DATA 160,0,177,253,240,9,32,
      168'BBMG
53134 DATA 207,81,253,240,2,145,253,
      230'BDCF
53142 DATA 253,208,2,230,254,165,253,
      197'BEKF
53150 DATA 45,208,231,165,254,197,46,
      208'BESE
53158 DATA 225,96,56,173,54,3,109,
      57'BACL
53166 DATA 3,109,59,3,141,54,3,
      162'BXOJ
53174 DATA 4,189,54,3,157,55,3,
      202'BXVI
53182 DATA 16,247,173,54,3,96,162,
      5'BYVI
53190 DATA 169,0,157,0,2,202,16,
      250'BYYH
53198 DATA 96,162,5,189,0,2,157,
      54'BXDO
53206 DATA 3,202,16,247,96,256'BTDE
```

END

BY MICHAEL LEIDEL

BASIC Macro Processor

for the Commodore 64, 128 and Plus/4



BOB CLARK

Using macros can improve programming development by saving time through less keying and program debugging.

In the mid 80's, many computer languages became inadequate to handle the more complex business and home computer applications. As a result, the high-level language macro processor appeared. The use of macros is not a new programming concept; the macro assembler has been around for many years. Now, however, macros are being used in high-level languages to improve the quality and speed of program development.

BASIC is a high-level language and a good candidate as a host language for macro processing. In this article, I refer to host as the program containing macros or the input program to the macro processor.

All BASIC's have the PRINT statement. But most do not have a statement that will center text before printing. If our BASIC does not have a CENTER command, then a series of statements must be used to accomplish this. A macro is a single command which represents many commands.

There are several points to consider on the subject of macros and BASIC Macro Processor. Writing macros involves creating new commands using parameter replacement, directives, and normal language statements. Macros are referenced in the host program by their name and parameters (or arguments). The macro processor scans the host program and its macros, expands macros according to parameters and directives found, and builds the new run program file.

Writing macros is very much like writing a program subroutine. However, there are some other considerations. The following are just some aspects to consider when writing macros for BASIC Macro Processor: parameters; directives (conditional, set, error, message, end, exit); macro comment lines; line numbering; and GOTOs, GOSUBs, and THEN.

Parameters (or arguments) are used in macros to represent a variable or literal which is replaced during macro processing or when the macro is expanded into the run program file. The replacement value for the parameter is given from the macro call in the host program. The following macro call uses three parameters.

["DISKIN,MYFILE,2,1"]

The macro name is DISKIN and the three parameters are MYFILE, 2 and 1. The macro writer will represent these parameters as ←01, ←02 and ←03, when writing a macro statement like the following.

OPEN ←02,8, ←02, "←01,S,R"

The macro processor will replace the ←01 with MYFILE and the ←02 with 2. This macro processor allows a maximum of 20 arguments per macro call. This is one of the ways that macros have intelligence over ordinary subroutines or external subroutines.

A conditional directive gives a macro the ability to include or exclude code depending on the value of its arguments. A boolean result determines whether code will be included (TRUE) or excluded (FALSE) from the run program file. In the following example,

```
10    !02 EQL "C64"
11    REM C-64 ROUTINE
12    ...
20    !END
```

Line 10 is the conditional directive. The value following the operator (EQL) must always be enclosed in quotes. In the above example, lines 11 through 19 will be included in the expanded program only if parameter 2 is equal to C64. Other operators that may be used are

- NEQ Not equal to
- LSS Less than
- GTR Greater than
- GEQ Greater than or equal to
- LEQ Less than or equal to

Note also line 20. Conditional directives must end with the END directive. The END directive tells the macro processor to begin including code following the END directive.

The EXIT directive stops all further processing of a macro. Upon reaching an EXIT directive, the macro processor stops processing from the macro file and resumes processing from the host program file. EXIT is usually used within a conditional directive structure.

Another directive is SET. This will assign a literal value to a parameter. It may be used in conjunction with the conditional directive. For example, in the following the second parameter is set to a value when it is detected to be NULL or empty.

```
!02 EQL ""
!SET 02 "255"
!END
```

Note that the value assigned to the parameter must always be enclosed in quotes. In the above example, it could be said that parameter 2 will default to the value 255 unless it is given a value in the macro call.

ERROR is another directive. It is used to terminate the macro processor run, usually because of an unacceptable value found in a parameter.

```
!04 GTR "10"
!ERROR "INVALID PARAM 4"
!END
```

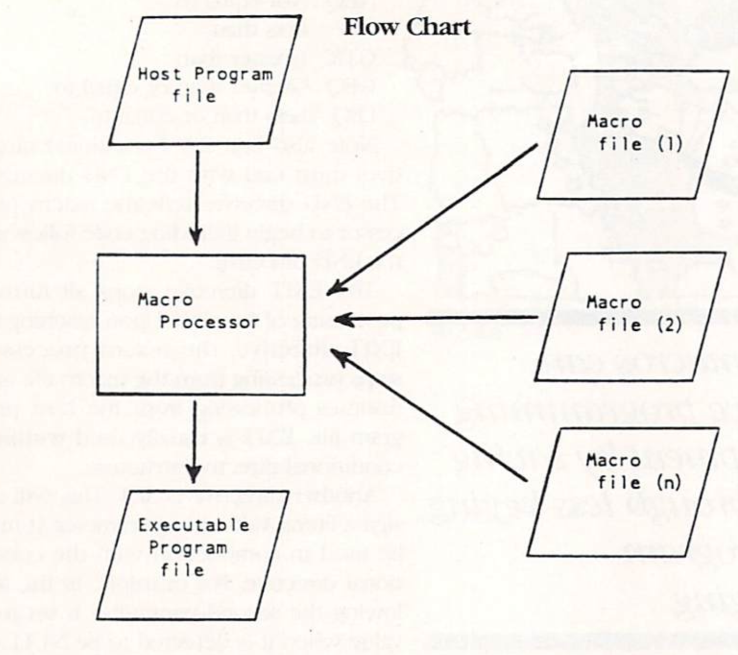
The above is directing the macro processor to abort and display a message if parameter 4 is greater than 10. Note that !ERROR will probably always be used with the conditional directive.

The MESSAGE directive is similar to ERROR. The difference is that when the macro processor encounters a MESSAGE directive, the message in quotes is displayed and processing continues. For example,

```
!03 EQL ""
!MESSAGE "NO ERR CHNL CHK for
DISKIN"
!END
```

Might be used to remind the programmer that code for reading the error channel will be excluded from the run program file.

Here are the six BASIC Macro Processor directive commands.



!01 LSS "10" Conditional
!END Code following is included
!EXIT Stops processing macro
!SET 01 "value"
Assigns value to parameter

!ERROR "message"
Aborts macro processor
!MESSAGE "message"
Displays message

Remember to always express argument numbers as two digits—02 not 2.

There is a special provision for making comments in macros. Any line beginning with a quotation mark is considered a comment and is NOT included in the generated run program file. It is suggested that these comment lines be used at the beginning of each macro to document the parameters it uses and its function.

Two final points on macro writing are line numbering and BASIC branch statements. The line numbers in a macro must always start with 1 and increment by 1. For branching to line numbers GOTO and GOSUB may be used, but not THEN. THEN may be used for an expression following a conditional IF, but not to branch to a line number.

Ordinarily, GOTO and GOSUB will be used to refer to lines in the macro. However, occasionally a branch outside the macro may be needed. When branching to line numbers that will be outside the macro, use the command GOTO# or GOSUB#. The pound-sign tells the macro processor not to compute a line

number within the macro but to use the literal number following the command as the line number.

A macro is referenced in the host program by its name and arguments. The proper syntax for calling a macro in the host program is ["macro-name,param-1,param2,...param20"] where [must be the first character after the line number.

Parameters may be left NULL or empty. In the following example, ["macro-name,p1,,p3,p4"], parameters 2, and 5 through 20 are NULL.

The macro processor scans the host program and its macros, expands macros according to parameters and directives found, and builds the new run program file. The macro is written and saved on disk independently of the host program. To use BASIC Macro Processor, both the macros and the host program(s) must be on the same disk. It is suggested to backup both macros and host programs on a separate disk before running BASIC Macro Processor.

To run BASIC Macro Processor just load it, type RUN, and enter the name of the host program. The run program file will have the same name as the host program file but with the extension .EXP (for expanded).

Using macros can improve program development by saving time through less keying and less program debugging. Writing macros can be fun and challenging. They can be as simple as a CENTER macro for centering text or as complex as a file handling routine for an application file used in many programs with different accessing needs.

Macro writing is a creative process like programming. Beyond creating an automated solution to a problem—game or whatever—macro writing produces a tool. Good macros can be used in more than one program and possibly more than one application. Often routines that have been used over and over again with only slight variations (if any) are good candidates to be made into macros.

Study the following macro, host program and expanded run program listings. Note how parameters are replaced and directives followed from the macro to be expanded program. Note how the macro calls are formatted in the host program. ■

Before typing this program, read "How to Enter Programs" and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

BASIC Macro Processor

```
10 REM *** MACRO PROCESSOR ***'BUNC
30 NØ$=CHR$(Ø):QUOTE$=CHR$(34):TRUE=1
   :FALSE=Ø:DIM PARM$(2Ø)'HPYN
40 GT$=CHR$(137):GS$=CHR$(141)'ERXG
50 PRINT"[CLEAR,DOWN,SPACE2,SHFT *7]
   [RVS]BASIC MACRO PROCESSOR[RVOFF]
   [SHFT *6,SPACE2,DOWN2]"'BAUT
60 PRINT"[DOWN] ENTER NAME OF HOST
   PROGRAM:":INPUT"[RIGHT7,DOWN]";P$
   :P$=LEFT$(P$,12)'EOJP
70 PRINT"[DOWN2]PROCESSING...":PRINT
   :MACRO=FALSE:CH=2'EQWN
80 OPEN 1,8,15:OPEN 2,8,2,"Ø
   :"+P$+",P,R":INPUT#1,E,E$
   :IF E=Ø THEN 1ØØ'IDKQ
90 CLOSE 2:CLOSE 1:PRINT E,E$:END'EJRJ
100 OPEN 3,8,3,"Ø:"+P$+".EXP,P,W"
    :INPUT#1,E,E$:IF E=Ø THEN 12Ø'HVYG
110 CLOSE 3:PRINT#1,"SØ:"+P$+".EXP"
    :INPUT#1,E,E$,E:PRINT E$,E
    :GOTO 1ØØ'HYQH
120 PRINT#3,CHR$(1);CHR$(8);:GOSUB 24Ø
    :GOSUB 24Ø'FSQD
130 REM **** MAIN PROCESSING LOGIC
    ****'BCKF
140 GOSUB 24Ø:D$=C$:GOSUB 24Ø
    :IF D$=NØ$AND C$=NØ$THEN 46Ø'IBCJ
150 GOSUB 26Ø:GOSUB 24Ø'CHEC
160 IF C$="[" THEN GOSUB 36Ø:L$=""
    :GOTO 14Ø'GMDH
170 IF C$="!" THEN IF MACRO THEN
    GOSUB 65Ø:GOTO 14Ø'HOSJ
180 IF EXCLUDE OR C$=QUOTE$ THEN IF
    MACRO THEN GOSUB 121Ø
    :GOTO 14Ø'IDEP
190 IF C$="[BACK ARROW]
    " THEN IF MACRO THEN GOSUB
    113Ø'GLKK
200 IF C$=GT$OR C$=GS$THEN IF MACRO
    THEN GOSUB 148Ø'ITRF
210 L$=L$+C$:IF C$=NØ$ THEN GOSUB 3ØØ
    :L$="":GOTO 14Ø'IWHG
220 GOSUB 24Ø:GOTO 19Ø'CHCA
230 REM * CLOSED SUBROUTINES FOLLOW
    *'BAKG
240 GET#CH,C$:IF C$=""THEN C$=NØ$'FOIG
250 RETURN'BAQB
260 GET#CH,LN$,HN$:IF LN$=""THEN
    LN$=NØ$'FVNK
270 IF HN$=""THEN HN$=NØ$'EJVH
280 IF MACRO THEN LM=LM+1
    :IF LM>255 THEN HM=HM+1:LM=Ø'KAHR
290 RETURN'BAQF
300 IF W=Ø THEN W=LEN(L$)+5
    :GOTO 32Ø'HMVE
310 W=LEN(L$)+4'DGGB
320 WT=WT+W:X=INT(WT/256)
    :HP$=CHR$(X+8):X=(WT-(X*256))
    :LP$=CHR$(X)'NQIS
330 IF MACRO THEN LN$=CHR$(LM)
    :HN$=CHR$(HM)'GUYJ
340 PRINT#3,LP$;HP$;LN$;HN$;L$;
    :RETURN'CWXH
350 REM *** OPEN MACRO FILE ***'BTDH
360 LF$="":L$="":IF MACRO THEN
    PRINT"CANNOT NEST MACROS"
    :GOTO 78Ø'GQYP
370 GOSUB 24Ø:IF C$<>CHR$(34)THEN
    PRINT#1,"I":PRINT"[DOWN]
    MISSING QUOTES":GOTO 78Ø'JRIR
380 GOSUB 24Ø:IF C$=","THEN GOSUB 49Ø
    :PRINT:GOTO 41Ø'HOQM
390 IF C$=QUOTE$ THEN GOSUB 24Ø
    :GOSUB 24Ø:GOTO 41Ø'GTMN
400 LF$=LF$+C$:PRINT C$;:GOTO 38Ø'EQCE
410 OPEN 5,8,5,"Ø:"+LF$+",P,R"
    :INPUT#1,E,E$:IF E=Ø THEN 43Ø'HWMJ
```

```
420 PRINT"[CLEAR]FILE ERROR";E;LF$;E$
    :GOTO 780'CNSH
430 MACRO=TRUE:CH=5:GOSUB 240
    :GOSUB 240'EVBI
440 LM=ASC(LN$):HM=ASC(HN$)
    :MB=HM*256+LM'HAON
450 L$=CHR$(143)+" "+LF$+" MACRO"+N0$
    :GOSUB 300:RETURN'ISUN
460 IF MACRO THEN CLOSE 5:MACRO=FALSE
    :CH=2:GOTO 130'GAPO
470 PRINT#3,CHR$(0);CHR$(0);:CLOSE 2
    :CLOSE 3:CLOSE 1'GQHM
480 END'BACG
490 REM COLLECT PARAMETERS'BRLM
500 FOR X=1 TO 20:PARM$(X)="":NEXT X
    :X=1'GSBH
510 IF X>20 THEN 580'DGHD
520 GOSUB 240'BDJC
530   IF C$=N0$THEN PRINT"[CLEAR,DOWN]
    MISSING QUOTE IN MACRO LINE"
    :GOTO 780'FJCO
540 IF C$=QUOTE$THEN 610'DLWH
550 IF C$="," THEN X=X+1:GOTO 510'GJOK
560 PARM$(X)=PARM$(X)+C$'CSNK
570 GOTO 510'BDFH
580 IF X>20 THEN PC=20:GOTO 610'FLHM
590 PC=X'BDGJ
600 REM 610 CHECKS FOR CLOSING ]
     NULL'BYAH
610 GOSUB 240:IF C$<>"]"THEN 630'FJFG
620 GOSUB 240:IF C$=N0$ THEN
    RETURN'FJPG
630 PRINT"[CLEAR] INVALID MACRO
    SYNTAX":GOTO 780'CEVL
640 REM ** HANDLE MACRO DIRECTIVE
    **'BYSK
650 D1$=""'BDYG
660 GOSUB 240'BDJH
670 IF C$=" "OR C$=N0$ THEN 700'FKIM
680 D1$=D1$+C$'CITK
690 GOTO 660'BDLK
700 IF ASC(D1$)=128 THEN EXCLUDE=FALSE
    :RETURN'GVEK
710 IF LEN(D1$)=2 THEN GOSUB 820
    :RETURN'GKCH
720 IF EXCLUDE THEN GOSUB 1220
    :RETURN'EMKI
730 IF D1$="ERR[CMDR A]"THEN 1240'DHYI
740 IF D1$="MESSAGE"THEN GOSUB 1310
    :RETURN'FIML
750 IF D1$="SET"  THEN GOSUB 1370
    :RETURN'FIOL
760 IF D1$="EXIT" OR D1$=CHR$(237)
    THEN CLOSE 5:MACRO=FALSE:CH=2
    :RETURN'KDUV
770 PRINT"[CLEAR,DOWN]INVALID MACRO
    DIRECTIVE ";D1$:GOTO 780'CIBS
780 REM *** ABORT ROUTINE ***'BSPO
790 IF MACRO THEN CLOSE 5'DGUN
800 CLOSE 2:CLOSE 3:CLOSE 1
    :PRINT"[DOWN2]UNABLE TO CONTINUE
    AT LINE ";'EHAN
810 PRINT(ASC(HN$)*256)+ASC(LN$)
    :END'GQKK
820 REM * HANDLE CONDITIONAL DIR
    *'BWLK
830 AGNBR=VAL(D1$)'CKCI
840 IF AGNBR<0 OR AGNBR>20 THEN
    PRINT"[CLEAR]INVALID ARG NUMBER ";
    D1$:GOTO 780'HVPV
850 D2$=""'BDAI
860 FOR X=1 TO 3:GOSUB 240:D2$=D2$+C$
    :NEXT X'HSCQ
870 GOSUB 240:GOSUB 240'CHCL
880 IF C$<>QUOTE$ THEN PRINT"[CLEAR]
    MISSING VALUE QUOTE ON !
    [BACK ARROW]#":GOTO 780'GMFY
890 D3$=""'BDBM
900 GOSUB 240:IF C$=N0$THEN PRINT"
    [CLEAR,DOWN]MISSING QUOTE ON !
    [BACK ARROW]#":GOTO 780'GNPP
910 IF C$<>QUOTE$THEN D3$=D3$+C$
    :GOTO 900'HUEN
920 GOSUB 240:REM GET LAST NULL'CPIK
930 IF C$<>N0$ THEN PRINT"[CLEAR]
    INVALID CONDITIONAL LINE IN MACRO"
    :GOTO 780'GJEV
940 IF D2$="EQL"THEN GOSUB 1010
    :RETURN'FIVM
950 IF D2$="LSS"THEN GOSUB 1030
    :RETURN'FION
960 IF D2$="GTR"THEN GOSUB 1050
    :RETURN'FILO
970 IF D2$="LEQ"THEN GOSUB 1070
    :RETURN'FICP
980 IF D2$="NEQ"THEN GOSUB 1090
    :RETURN'FIGQ
990 IF D2$="GEQ"THEN GOSUB 1110
    :RETURN'FIRR
1000 PRINT"[CLEAR]INVALID CONDITIONAL
     OPERATOR ";D2$:GOTO 780'CIHE
1010 IF PARM$(AGNBR)=D3$ THEN
     EXCLUDE=0:RETURN'FYPD
1020 EXCLUDE=1:RETURN'CJFX
1030 IF PARM$(AGNBR)<D3$ THEN
     EXCLUDE=0:RETURN'FYQF
1040 EXCLUDE=1:RETURN'CJFA
1050 IF PARM$(AGNBR)>D3$ THEN
     EXCLUDE=0:RETURN'FYOH
1060 EXCLUDE=1:RETURN'CJFC
1070 IF PARM$(AGNBR)<=D3$ THEN
     EXCLUDE=0:RETURN'GYTJ
1080 EXCLUDE=1:RETURN'CJFE
1090 IF PARM$(AGNBR)<>D3$ THEN
     EXCLUDE=0:RETURN'GYSL
1100 EXCLUDE=1:RETURN'CJFW
1110 IF PARM$(AGNBR)=>D3$ THEN
     EXCLUDE=0:RETURN'GYRE
1120 EXCLUDE=1:RETURN'CJFY
1130 REM * HANDLE PARAMETER
     REPLACEMENT'BCDF
1140 GOSUB 240:D1$=C$:GOSUB 240
     :D1$=D1$+C$'FWEF
1150 AGNBR=VAL(D1$)'CKCC
1160 IF AGNBR<0 OR AGNBR>20 THEN
```

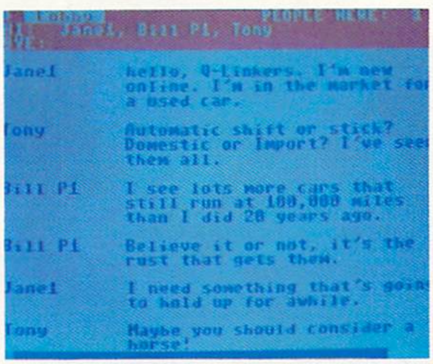
BASIC MACRO PROCESSOR

```
     PRINT"[CLEAR,DOWN]
     INVALID ARGUMENT ";D1$
     :GOTO 78Ø'HVQO
117Ø L$=L$+PARM$(AGNBR)'CQYG
118Ø GOSUB 24Ø:REM GET BYTE AFTER
     [BACK ARROW]ARG'CFJL
119Ø RETURN'BAQD
12ØØ REM * HANDLE MACRO COMMENT *'BUUB
121Ø GOSUB 24Ø'BDJW
122Ø IF C$<>NØ$ THEN 121Ø'EJXB
123Ø RETURN'BAQX
124Ø REM * HANDLE ERROR ABORT *'BSTE
125Ø E$="":GOSUB 24Ø:REM READ
     QUOTE'DQJF
126Ø GOSUB 24Ø:IF C$=QUOTE$THEN
     129Ø'EQMH
127Ø IF C$=NØ$THEN 13ØØ'DJBF
128Ø E$=E$+C$:GOTO 126Ø'DLHH
129Ø GOSUB 24Ø:REM READ NULL'CMQI
13ØØ PRINT"[CLEAR,DOWN]";E$
     :GOTO 78Ø'CHRY
131Ø REM * HANDLE WARNING MSG *'BSOC
132Ø PRINT"M> ";'BBDY
133Ø GOSUB 24Ø:REM GET QUOTE'CMOD
134Ø GOSUB 24Ø:IF C$=QUOTE$THEN GOSUB
     24Ø:PRINT:RETURN'HRYI
135Ø IF C$=NØ$ THEN PRINT:RETURN'FGCF
136Ø PRINT C$;:GOTO 134Ø'CIWE
137Ø REM * HANDLE SET DIRECTIVE *'BUAJ
138Ø GOSUB 24Ø:D2$=C$:GOSUB 24Ø
     :D2$=D2$+C$'FWHL
139Ø ARGNBR=VAL(D2$)'CLEJ
14ØØ IF ARGNBR<1 OR ARGNBR>2Ø THEN
     PRINT"[CLEAR,DOWN]
     INVALID ARG NUMBER ";D2$
     :GOTO 78Ø'HXXM
141Ø GOSUB 24Ø:GOSUB 24Ø'CHCA
1415 IF C$<>QUOTE$THEN PRINT"[CLEAR,
     DOWN]MISSING QUOTE ON !SET"
     :GOTO 78Ø'GMSQ
142Ø D3$=""'BDBA
143Ø GOSUB 24Ø:IF C$=NØ$THEN PRINT"
     [CLEAR,DOWN]MISSING QUOTE ON
     !SET":GOTO 78Ø'GNWM
144Ø IF C$<>QUOTE$THEN D3$=D3$+C$
     :GOTO 143Ø'HVAK
145Ø GOSUB 24Ø:REM GET LAST NULL'CPIH
146Ø PARM$(ARGNBR)=D3$'BQXH
147Ø RETURN'BAQE
148Ø REM HANDLE MACRO BRANCH'BRPK
149Ø B$="":L$=L$+C$'DJTJ
15ØØ GOSUB 24Ø:IF C$=" "THEN 15ØØ'EKQC
151Ø IF C$="#"THEN GOSUB 24Ø
     :RETURN'FGOD
152Ø B$=B$+C$:GOSUB 24Ø'DKKD
153Ø IF C$=":"OR C$=NØ$OR C$=",
     "THEN 155Ø'HNCI
154Ø GOTO 152Ø'BEFD
155Ø BO=VAL(B$):B$=STR$(MB+BO)'FPOJ
156Ø IF C$=NØ$OR C$=":"THEN L$=L$+B$
     :RETURN'IOPM
157Ø L$=L$+B$+C$:B$="":GOSUB 24Ø
     :GOTO 153Ø'GUMM
```

END

JOIN THOUSANDS OF OTHER COMMODORE OWNERS WHO ARE CONNECTING TO QUANTUMLINK TO:

- Get computer help
- Access software
- Play games
- Learn programming
- Talk to Commodore experts
- Win prizes
- Make travel reservations
- Chat live
- Send mail
- Preview "hit" software
- Use an encyclopedia
- Tap into a newswire
- Meet friends
- Join clubs
- Order products
- Save time
- Save money
- Have a lot of fun



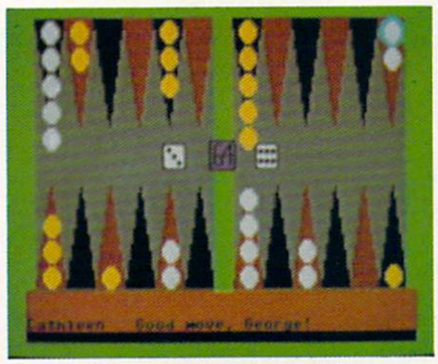
Find people who share your interests and let your fingers do the talking. Chat live. Send mail.

THE ONLY OFFICIAL COMMODORE ONLINE SERVICE

QuantumLink is the only network actively supported by Commodore. And because QuantumLink is available only to Commodore owners, the services are carefully selected to meet your needs. You'll make better use of your computer, plus have lots of fun.

NO OTHER ONLINE SERVICE IS EASIER TO USE

You will find QuantumLink very friendly. You don't have to memorize any complicated commands—all you do is select options from the colorful, easy-to-read menus.



Play games with people located all across the country. Compete in tournaments and win prizes.

THE LOWEST-PRICED FULL-SERVICE NETWORK

For only $9.95 a month, you get UNLIMITED access to a wide range of base services including up-to-the-minute news, the encyclopedia, the Commodore Hotline, and much more.

Your first hour of special "Plus" services, including chat and downloading, is also FREE every month. After your first hour, "Plus" services cost only 6 cents per minute.

THERE IS ABSOLUTELY NO OBLIGATION

If at any time you are not completely satisfied with QuantumLink, just call Customer Service to cancel your membership.

Special 1 Month FREE offer—you get your second month of base services FREE when you pay for the first! (A $9.95 value!)



Access thousands of programs including previews of "hit" software. Submit your own programs.

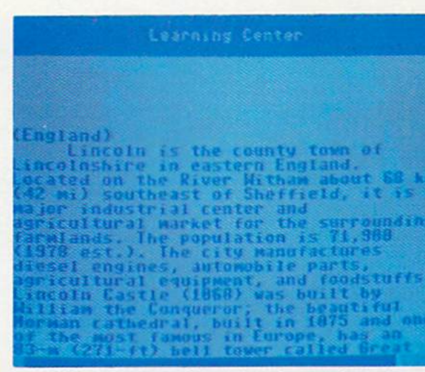
PRICED AT ONLY $9.95!*

The QuantumLink Starter Kit includes everything you need: telecommunications software, a user guide, and a list of local access numbers.

AVAILABLE NOW

You can get QuantumLink right now from your local Commodore retailer. If QuantumLink is not in stock, just order by phone.

Call toll-free: 1-800-392-8200—Ask for Dept. 67.



Utilize a constantly updated encyclopedia. Get online tutoring. Read the latest news.

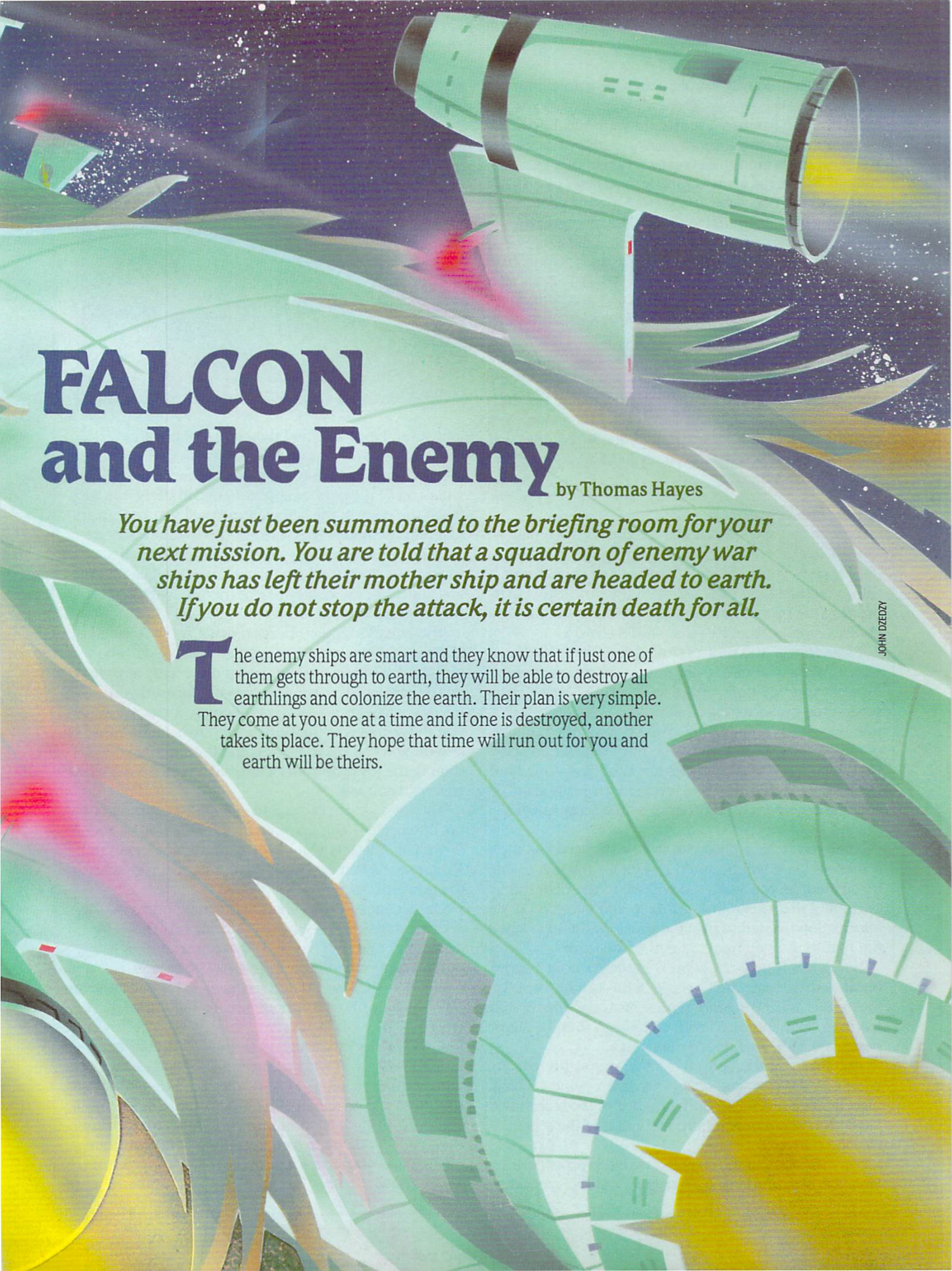
*Suggested retail price. Dealer price may vary.

Dealers/Distributors: for ordering information, call Ed Rothstein at Quantum Computer Services: (703) 448-8700.

QuantumLink SM
The Commodore® Connection

8620 Westwood Center Drive • Vienna, VA 22180 • 703-883-0788





FALCON and the Enemy

by Thomas Hayes

You have just been summoned to the briefing room for your next mission. You are told that a squadron of enemy war ships has left their mother ship and are headed to earth. If you do not stop the attack, it is certain death for all.

JOHN DZEDZY

The enemy ships are smart and they know that if just one of them gets through to earth, they will be able to destroy all earthlings and colonize the earth. Their plan is very simple. They come at you one at a time and if one is destroyed, another takes its place. They hope that time will run out for you and earth will be theirs.

Your plan is equally simple. You have a time limit in which you must destroy all enemy ships to save the earth from destruction. You rush to your new Falcon XY-8 computer-guided freedom flier. You strap yourself in and push the button to turn on your computer guidance system. To the right, the display informs you of systems ready and the computer jets you off to start your mission. All systems are handled by the computer except the firing of your laser-generated missiles.

It's your job to shoot down enemy ships. Suddenly, the first ship is in sight. You fire. The special effects alert you to the fact that this is not just a game—it's life and death out here and the only one you can depend on is yourself.

After your first hit, you glance over at the time you have and rush in vain to destroy the enemy before they destroy you. Do you have the guts, the ability, and the stamina needed to fight this life and death mission—not only for yourself but for the friends and loved ones you leave behind? If you dare, read on and find out what is needed to be a hero.

Game Description

Type in and save Falcon and the Enemy, then plug a joystick into port two and run the program. The first display gives you a choice of time and the number of ships you think you can destroy in the time allowed. By pressing the proper function key, you have a number of choices ranging from F1 (easy) to F8 (hard).

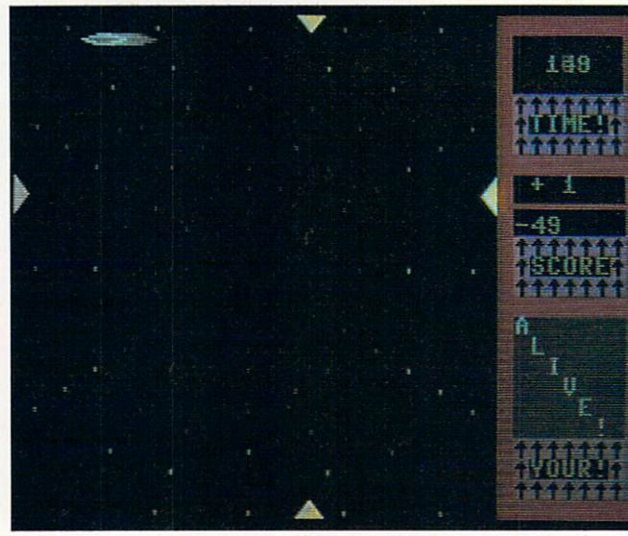
F1 gives you 400 time-cycles to destroy 50 ships, which means you have eight time-cycles to every ship. F2 gives you 525 time- cycles to destroy 75 ships. That's seven time-cycles for every ship. As you increase the number on the function key, you decrease the number of time-cycles for each ship. The most difficult choice is F8, which has 225 time-cycles for 225 ships—one time-cycle for each ship. The time-cycles don't really stand for any real time period, and they're a little longer than a second.

After you make a choice, it will be displayed on the screen and you will be asked to wait for the universe to be created. As you wait, you will see to the right the portion of the screen that will keep you informed of the information needed to play the game. At the top right is the time remaining until the end of the game.

Below the time is the score. I hesitate to call it score because there really isn't a score in this game. You either live or die. What the score represents is the enemy ships left to destroy, indicated with a plus sign, and the ships destroyed, indicated with a minus sign.

And finally, at the bottom right is your status, which will flash ALIVE on and off as long as the game is being played. After several seconds, the rest of the screen is filled with stars coming from the inside of the screen moving outward.

The 3D effect of the stars is a two-step procedure. First, the machine-language program inserts a period



into several locations and then replaces it with a space character moving outward as it runs. Secondly, several colors of gray and white are poked into various locations which make the periods appear to become larger as they reach the edge of the screen. I can't tell you how many different methods I tried before I decided on this one.

There are four yellow triangular shapes which indicate the location of your laser missiles. The missiles will appear in the center where the yellow triangular shapes point like the crosshairs on a gun scope.

The enemy's ship is always moving and hovering just in front of your ship. By using the joystick you can position the crosshairs to shoot down the enemy anywhere on the screen. By moving the joystick forward you move the side triangles up and by pulling the joystick toward you the triangles at the side move down. In a similar fashion you can move the top and bottom triangles by shifting the joystick from side to side to move the triangles right and left.

The joystick can be read in eight different positions: up, down, left, right, up-right, up-left, down-right, and down-left. This lets you move all four triangles at the same time. When you push the fire button you unleash the powerful laser missiles.

On occasion, it may seem that you hit only one ship, yet two ships were registered as being hit. This is because after one ship has been destroyed another ship is put in a random position which may be close to the area where you shot last. After a few games this will all seem second nature, so if it sounds strange right now, don't worry.

If you are lucky enough to destroy all the ships before time runs out, you will be congratulated and asked if you wish another mission. If you do, just hit the spacebar and start from the beginning. If you don't make it alive, well, you don't make it alive—there's not much to say. However, your mothership may be able to complete your mission and you may get a second chance—you just never know. C

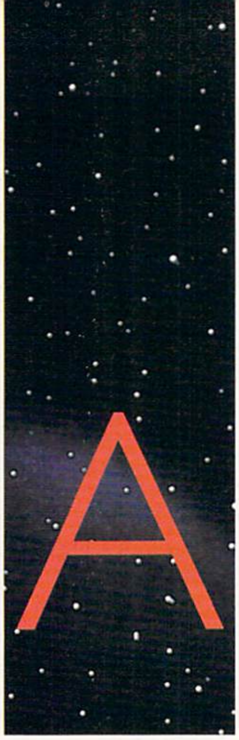
Before typing this program, read "How to Enter Programs" and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Falcon and the Enemy

for the Commodore 64

```
5 POKE 53280,0:POKE 53281,0
  :PRINT"[CLEAR,GREEN]"'DQTI
10 GOSUB 55:FOR X=49152 TO 50005
   :READ A:POKE X,A:NEXT'HVDF
11 PRINT"[CLEAR,RVS] PRESS FOR
   :[RVOFF,SPACE6]SHIPS[SPACE6]
   TIME!"'BAPH
12 PRINT"[DOWN2](EASY)[RVS] F-1
   :[RVOFF,SPACE7]50[SPACE9]400"'BAHF
13 PRINT"[DOWN,SPACE6,RVS] F-2
   :[RVOFF,SPACE7]75[SPACE9]525"'BAFG
14 PRINT"[DOWN,SPACE6,RVS] F-3
   :[RVOFF,SPACE7]100[SPACE8]600"'BAGH
15 PRINT"[DOWN,SPACE6,RVS] F-4
   :[RVOFF,SPACE7]125[SPACE8]625"'BAVI
16 PRINT"[DOWN,SPACE6,RVS] F-5
   :[RVOFF,SPACE7]150[SPACE8]600"'BANJ
17 PRINT"[DOWN,SPACE6,RVS] F-6
   :[RVOFF,SPACE7]175[SPACE8]525"'BACK
18 PRINT"[DOWN,SPACE6,RVS] F-7
   :[RVOFF,SPACE7]200[SPACE8]400"'BAJL
19 PRINT"[DOWN](HARD)[RVS] F-8
   :[RVOFF,SPACE7]225[SPACE8]225"'BAEM
20 PRINT"[RED,SHFT *40,GREEN,DOWN]
   ENTER[RVS] F-1:[RVOFF]TO[RVS] F-8
   :[RVOFF]"'BALJ
21 PRINT"[DOWN]YOUR CHOICE IS
   :[SPACE2]";'BBPE
22 GET A$:IF A$=""THEN 22'EHLD
23 A=ASC(A$):IF A<133 OR A>140 THEN
   PRINT"[RVS] PICK AGAIN [RVOFF,
   LEFT12]";:GOTO 22'JSXV
24 IF A=133 THEN SH=50:PX=400
   :PRINT"[RVS] F-1:[RVOFF,SPACE7]"
   :GOTO 32'HSYN
25 IF A=137 THEN SH=75:PX=525
   :PRINT"[RVS] F-2:[RVOFF,SPACE7]"
   :GOTO 32'HSSO
26 IF A=134 THEN SH=100:PX=600
   :PRINT"[RVS] F-3:[RVOFF,SPACE7]"
   :GOTO 32'HTWP
27 IF A=138 THEN SH=125:PX=625
   :PRINT"[RVS] F-4:[RVOFF,SPACE7]"
   :GOTO 32'HTQQ
28 IF A=135 THEN SH=150:PX=600
   :PRINT"[RVS] F-5:[RVOFF,SPACE7]"
   :GOTO 32'HTFR
29 IF A=139 THEN SH=175:PX=525
   :PRINT"[RVS] F-6:[RVOFF,SPACE7]"
   :GOTO 32'HTXS
30 IF A=136 THEN SH=200:PX=400
   :PRINT"[RVS] F-7:[RVOFF,SPACE7]"
   :GOTO 32'HTCK
31 IF A=140 THEN SH=225:PX=225
   :PRINT"[RVS] F-8:[RVOFF,SPACE7]
   "'GQUK
32 POKE 767,SH:FOR D=0 TO 700
   :NEXT'FNRG
33 PRINT"PLEASE WAIT: THE UNIVERSE IS
   BEING MADE";:FOR D=0 TO 1400
   :NEXT'FJCR
50 PRINT"[CLEAR]":F=PEEK(53278)
   :POKE 49989,194:POKE 49994,63
   :GOTO 100'GHYL
55 PRINT"[RVS] PROGRAM[SPACE4,DOWN,
   LEFT12] WRITTEN BY:[RVOFF]";'BBCU
56 PRINT"[SPACE3]THOMAS HAYES"'BAFL
60 RETURN'BAQC
100 GOSUB 4000:SYS 49987:POKE V+21,143
    :FOR L=0 TO PX:SYS XX
    :POKE 2,(RND(1)*15)+1'MMGK
102 PRINT AL$O$;:SYS XX+6:FOR S=0 TO 5
    :SYS XX:SYS XX+12:SYS XX+12
    :SYS XX+12'NHLN
110 POKE 821,(RND(1)*181)+58
    :POKE 822,(RND(1)*159)+60:SYS XX+6
    :SYS XX+12'MNYL
115 QO=PEEK(767):IF QO=0 OR QO>225
    THEN 300'HTKK
130 SYS XX:PRINT BL$"+"PEEK(767);
    :SYS XX+6:NEXT S'GSDG
140 SYS XX:PRINT"[LEFT5,DOWN2]
    "PEEK(767)-SH;:SYS XX+6:SYS XX
    :PRINT"[LEFT] ";G$PX-L;'JBUP
150 SYS XX+6:SYS XX:PRINT AL$I$O$;
    :SYS XX+6:NEXT L:GOTO 1000'IBWL
300 GOSUB 2000:'BFLX
400 PRINT"[HOME,DOWN,GREEN,RVS]
    ********************************
    ********";'BBLG
401 PRINT"[RED,RVS] YOU HAVE SAVED
    THE LIVES OF MILLIONS [GREEN]*";
    'BBRK
402 BS$="*[SPACE38]*":PRINT
    BS$BS$BS$BS$BS$;'CUSL
403 PRINT"****************************
    ************[RVOFF]";'BBFJ
404 PRINT"[HOME,DOWN5,RIGHT13]
    CONGRATULATIONS";:GOSUB 3000
    :PRINT'DHHL
405 PRINT"[DOWN6,RIGHT6,SPACE5]
    FOR ANOTHER MISSION[SPACE4]"'BAFL
406 PRINT"[DOWN2,RIGHT7,RVS,RED,^27]
    "'BAJP
407 PRINT"[RIGHT7,RVS,^,RVOFF,GREEN,
    SPACE5]PRESS SPACE BAR[SPACE5,RED,
    RVS,^]"'BAJN
408 PRINT"[RIGHT7,RVS,^27,GREEN]";
    :GOTO 1060'CGST
1000 GOSUB 2000:'BFLT
1010 RI$="[HOME,DOWN5,RIGHT30,CYAN]"
     :R$="[LEFT9,DOWN]"'CGXG
1020 RP$="[SHFT +,DOWN,LEFT4,SHFT U,
     SHFT *2,CMDR E,SHFT *2,SHFT I,
     DOWN,LEFT8,SHFT U,SHFT K,SPACE5,
     SHFT J,SHFT I]":RP$=RP$+R$:'DNUR
1030 RP$=RP$+"[SHFT -] [PURPLE]R.I.P.
     [CYAN,SHFT -]"+R$+"[SHFT -,
     SPACE7,SHFT -]"+R$+"[SHFT -,
     SPACE7,SHFT -]"+R$+"[SHFT -,
     L. RED]YOUR[CYAN,SPACE3,SHFT -]
     "'IMIS
1040 RP$=RP$+R$+"[SHFT -,L. RED]GRAVE
     [CYAN,SPACE2,SHFT -]"+R$+"
```

Continued on pg. 124



BY M. GARAMSZEGHY

A Beginner's Guide to Burst Mode

De-mystify the Powerful Set of Burst Commands

The disk operating system (DOS) of the Commodore 1571 supports an extended set of commands which enables the drive to create, read and write disks in a wide variety of formats with relative ease. To the novice, the description of these commands, collectively called burst mode, is at best cryptic.

Many 1571 owners have used burst mode in its various forms without even realizing it. Burst mode is used by the Commodore 128's CP/M to access a variety of foreign disk formats with the 1571 drive. A form of burst mode is also used by BASIC 7.0 commands such as BLOAD when the 128 is connected to a 1571.

Many, but not all, of the burst mode commands have analogous commands in standard Commodore DOS. Why then is there a need to duplicate them in burst mode? The answer is twofold: speed and versatility. The data transfer rate of a 1541 drive (or a 1571 in 1541 mode) is about 350 bytes per second. The 1571 in fast mode can read about 1,200 bytes per second. With burst mode, data can be read at the blistering rate of 2,500 or more bytes per second! In fact, the ultimate read speed is not limited by the hardware. It is controlled by the efficiency of the driving software. (That is, burst mode can, in theory, transfer data as fast as you can send it in machine language.)

In addition, burst mode allows you to read and write disks formatted in the industry standard MFM (maximum (or modified) frequency modulation) format as well as Commodore GCR (group coded recording) format. The MFM format is used by most CP/M


POWER
DRIVE
commodore 1571
DISK DRIVE

computers such as Kaypro, Epson and Osborne, as well as MS-DOS (IBM-PC and compatible) and TRS-80 machines. Apple II-type computers use a modified form of GCR which is not compatible with either MFM or Commodore GCR.

Burst mode commands are accessed by sending a specific set of characters through the disk command channel, just like the usual Commodore DOS commands such as n0:, s0: and b-p:. Because the first two characters of the command string for all burst mode commands are u0, burst commands are often referred to as the u0 commands. Burst mode can be accessed by a BASIC statement such as OPEN 15,8,15,"u0"+additional characters or the machine language equivalent. The additional characters depend on the desired function and its options. The first additional character selects the function and perhaps one or two primary options. The remaining characters select secondary options. Not all of the secondary option characters need to be sent in some cases because default values can often be assumed.

Table 1 gives a summary of some of the more common burst mode commands along with the byte sequence re-

Table 1
Burst Mode Command Summary

FUNCTION	BYTE SEQUENCE u0 + chr$( )	BURST INPUT	BURST OUTPUT
INQUIRE DISK: reset drive and log in MFM or GCR disk before a read or write			
MFM disk side 0 or GCR disk	4	(none)	one status byte
MFM disk side 1	20		(same as above)
QUERY DISK FORMAT: analyze disk format (GCR or MFM-sector size, sector/track)			
Side 0, track 0	10	(none)	one status byte then: disk or unreadable format or else: another status byte, number of sectors on track, logical track#, min sector#, max sec.# hard sector interleave
Side 1, track 0	26		(same as above)
Side 0, track n	138,n		(same as above)
Side 1, track n	154,n		(same as above)
READ SECTOR: (data transfer via burst mode)			
MFM disk side 0 or GCR disk (either side)	64,track#,sector, # of sectors	(none)	for each sector read: one status byte then data bytes
MFM disk side 1	80,track#,sector#, # of sectors		(same as above)
READ SECTOR: (data transfer via memory-read)			
MFM disk side 0 or GCR disk (either side)	192,track#,sector#,1		(none)
MFM disk side 1	208,track#,sector#,1		(same as above)
FASTLOAD: read an entire GCR file			
SEQ file	159,filename character bytes... + "*"	(none)	for each sector read: one status byte then 254 data bytes last sector: status byte = 31, next byte = number of bytes left, then rest of data bytes.
PRG file	31,filename char bytes... + "*"		(same as above)
WRITE SECTOR: (data transfer via burst mode)			
MFM disk side 0 or GCR disk (either side)	66,track#,sector#, # of sectors	data bytes	one status byte after each sector transferred
MFM disk side 1	82,track#,sector#, # of sectors		(same as above)
WRITE SECTOR: (data transfer via memory-writes)			
MFM disk side 0 or GCR disk (either side)	194,track#,sector#,1		(none)
MFM disk side 1	210,track#,sector#,1		(same as above)
SET SECTOR INTERLEAVE: (for multi-sector read and write)			
Set interleave	8,interleave	(none)	(none)
Read last setting	136	(none)	last setting
FORMAT DISK:			
MFM single sided	70,129,0,sector size (0-128 bytes/sector, 1 = 256, 2 = 512 or 3 = 1024), last track# (default 39), # of sectors per track, starting track# (default 0), track offset (default 0), fill byte (default hex e5)	(none)	(none)
MFM double sided	first byte = 102 then rest same as above		
GCR disk double sided no directory or BAM	6,0,ID byte#1, ID byte#2	(none)	(none)
INQUIRE STATUS: check drive status or load status register			
Log in disk with new status	76,new status	(none)	(none)
Check last status	140	(none)	status from last I/0
Check if disk was logged	204	(none)	old status if logged states error code 13

Table 2
Important 1571 Memory Locations

Address Hex	Dec	Function
24-29	36-41	ID field of last MFM sector accessed
24	36	track #
25	37	0 = side 0, 1 side 1
26	38	sector#
27	39	sector size: 0 = 128 bytes, 1 = 256, 2 = 256, 3 = 512, 4 = 1024
28-29	40-41	CRC (checksum)
37	55	Bus status byte: bit 0 1 = "file has only one sector" 3 inverse status of clock line 6 1 = 1571 bus mode, 0 = 1541 bus mode 7 1-1571 operating mode (2 MHz) 0-1541 operating mode (1 MHz)
3B	59	Last burst mode command number
3C	60	MFM disk sector interleave
5E	94	Burst status byte: bit 0-3 current error message number returned by job loop 7 1 = MFM, 0 = GCR
5F	95	Current job code
60	96	Smallest sector number on track
61	97	Largest sector number on track
67	103	Current track number
97	151	Number of MFM sectors per track
200-229	512-553	Input buffer for command strings from computer
274	628	Length of command string in input buffer
300-7FF	768-2047	RAM buffers: $300 is the start of buffer for MFM sector reads/writes
1800-180F	6144-6159	6522 VIA# 1: Controls 1541 bus and electronics
1C00-1C0F	7168 7177	6522 VIA# 2: Controls GCR recording electronics, motor, etc.
2000-2003	8192 8195	WD 1770: Controls MFM recording electronics
4000-400F	16384-16399	6526 CIA# 1: Controls 1571 fast bus mode
8000-FFFF	32768-65535	ROM 32K disk operating system

quired to access them. All byte values are in decimal. It should be noted that error checking is not performed on any of the parameters before they are passed to the 1571. This must be done by the calling program before the bytes are sent. Where no burst data input or output are requested, the function can be called entirely from BASIC with no need to resort to high-speed machine language. By using special options, even burst mode input and output can be avoided, but at the price of reduced speed.

Some of the command bytes have been simplified from what appears in the 1571 manual by selecting the most frequently used options. Other options may be available for some of the commands. A detailed bit-by-bit description of each command string can be found in the 1571 manual.

Command Summary

Before data can be read from or written to a disk using burst mode, it is necessary to log in the disk. This is somewhat equivalent to the normal Commodore DOS I0 command. After logging in, data can be read or written at will. If you remove the disk and change it, or even re-insert the same one, you will have to re-log the disk. The disk can be logged in by one of several methods. The simplest is to use the INQUIRE DISK command. The disk drive will send back a single status byte to the 128 representing some data on the disk format and the current error status.

If more detailed information is required about the format of the disk (such as number of sectors per track or sector numbering system), then the QUERY DISK FORMAT command should be used. This command can be used to analyze the format of a specific track on the disk, while the INQUIRE DISK only looks at the first track on the disk. The status bytes returned by these log-in commands need not be read by the calling program on the 128 and can be ignored if you just want to log the disk and are not interested in any errors that may have occurred.

Data can be read from the disk by one of two methods. The first method is the FAST LOAD command. Unlike the other burst mode commands, FAST LOAD acts on an entire file and will work only on normal Commodore DOS GCR files. The routine will work, however, with either PRG or SEQ type files. I have found that for consistent performance the DOS wild card character (*) should be appended to the filename being read. (FAST LOAD will not always recognize a legitimate filename unless it ends with a *.) FAST LOAD will return 255 bytes (1 status byte allowed by 254 data bytes) for each sector read. Remember that normal Commodore DOS sectors contain only 254 data bytes because the first two bytes are used to link to the next sector in the file. Subsequent sectors in a file are linked automatically.

The second method (READ SECTOR) is similar to the Commodore DOS Block-Read: (b-r: or ul:) command. The command will work with either GCR or MFM disks. For GCR disks, the command returns 257 bytes (1 status byte followed by 256 data bytes) for each sector read. MFM disks can have sector lengths of 128, 256, 512 or 1,024 bytes. The number of bytes returned is always one status byte followed by data bytes for one sector.

Unlike DOS Block-Read:, the burst mode command can be used to read more than one sector, up to one whole track at a time. For multi-sector reads, the pattern of one status byte followed by one sector of data bytes is repeated for the specified number of sectors. If you specify more sectors to read than are contained on one track, the data from the same track will be read over again until the specified number of sectors have been sent.

The burst WRITE SECTOR command is somewhat analogous to the standard Commodore DOS Block-Write: (b-w: or u2:) command. Unfortunately, there is no FAST SAVE command corresponding to the burst mode FAST LOAD command which would allow you to write an entire file in burst mode. Similar to most other burst mode commands, the write command will work with either MFM or GCR disks.

The burst write is also faster than normal KERNAL-controlled writing to the 1571, although the difference in speed is not as great as the difference in reading speeds. The average speed for a burst write using 256 byte sectors is about 600 bytes per second. The corresponding figure in normal 1571 mode is about 400 bytes per second and in 1541 mode it is about 300 bytes per second. In 1571 and burst modes, the write speeds are a factor of 3 to 5 slower than the corresending read speeds.

Similar to the burst READ SECTOR, burst write can also be used to write multiple sectors in succession (up to one track's worth). It should be noted that if you specify more sectors to write than are in one track, the command will keep writing data over the same track until the specified number of sectors have been written. This could result in a hopelessly corrupted disk if you are not careful.

One good application of multi-sector burst reads and writes is a disk copier where a large number of sectors are read from a source disk then written to the corresponding locations on a new disk.

Both the READ SECTOR and WRITE SECTOR commands have several very useful options, one of which allows the transfer of data through the normal disk command channel in either BASIC or machine language using the Commodore DOS m-r (memory read) and m-w (memory write) commands to directly access the data buffer in the 1571.

The order that multiple sectors are read or written with the above commands can be changed using the SET SECTOR INTERLEAVE command. The default interleave 1 corresponds to contiguous sectors. That is, if you started at track 10, sector 1 and read 3 sectors, you would read sector 1 then 2 then 3. If you changed the interleave to 3, you would read sector 1 then 4 then 7. Standard Commodore DOS has an interleave of 10 (except for the directory, track 18, which has an interleave of 3). The 128's CP/M has an interleave of 5, while most MFM disks do not use software interleaves.

There are some optional parameters in the burst mode FORMAT command which allow you to partially format a disk, format different areas of the disk in a different manner, change the sector slew rate, and change the fill byte.

Many of the commands will return a value called the status byte. This byte has the same structure in most cases. A bit-by-bit description is provided in the 1571 disk drive manual. In general, the low order 4 bits represent the status of the disk controller. A decimal value of 0 or 1 for these four bits indicates that everything is fine. Any other value indicates a controller error as listed in the 1571 manual.

The high order 4 bits are used for MFM disks. If bit 7 is set, then the disk is MFM format. The remaining bits give data on the number of bytes per sector. The status can be checked at any time using the INQUIRE STATUS command.

Continued on pg. 114

Interfacing Commodore's User Port PART 1

Connecting a computer to an external device or circuit is known as interfacing. Interfacing controls, reads or exchanges data from your computer to an external device, circuit or another computer. Applications range from controlling appliances and electronic devices to computer security systems, robotics and bio-feedback devices. Or have the computer monitor and react to light, sound, temperature, pressure and vibrations. The applications are limited only by your imagination.

Commodore engineers have been very generous in their design of computers, allowing users access to various input/output (I/O) ports. This is just one reason for Commodore's immense popularity.

Our concentration will focus on the user port located at the back of the VIC 20, Commodore 64 and 128. These computers all have similar user ports (see Diagram A) that function basically the same.

We will be accessing port B of the user port labeled PB0 through PB7. This is an eight-bit parallel port. Each bit on the port is bi-directional and is programable as either an input or output bit. Each bit on the port can also be programmed independently from all the other bits.

6526 Chip

Commodore computers use an integrated circuit chip between the central processing unit (CPU) and the I/O ports. The 64 and 128 use a 6526 CIA (complex interface adaptor) chip. The VIC 20 uses a 6522 VIA (versatile interface adaptor) chip. In order to avoid confusion, further descriptions of the 6526 CIA chip will be the only one given, and should be assumed to be the same for the 6522 chip unless otherwise noted.

All input and output functions are transmitted by the CIA chip to the micro-processor CPU. Each 6526 CIA chip contains two parallel eight-bit I/O ports, two 16-bit counter/timers, clock and a serial shift register. The chip is responsible for the 60 HZ interupt routine, keyboard scanning, game port reading and serialized data input. We will learn to use the timers, interupt routines serialized data input later on in the series. For now we will concentrate on basic functions.

All accessing of our user port is through the 6526 chip. To access this chip we must be able to set and read various bits on the chip registers. This isn't as difficult as it may sound, but it does require a basic understanding of the binary number system.

Binary

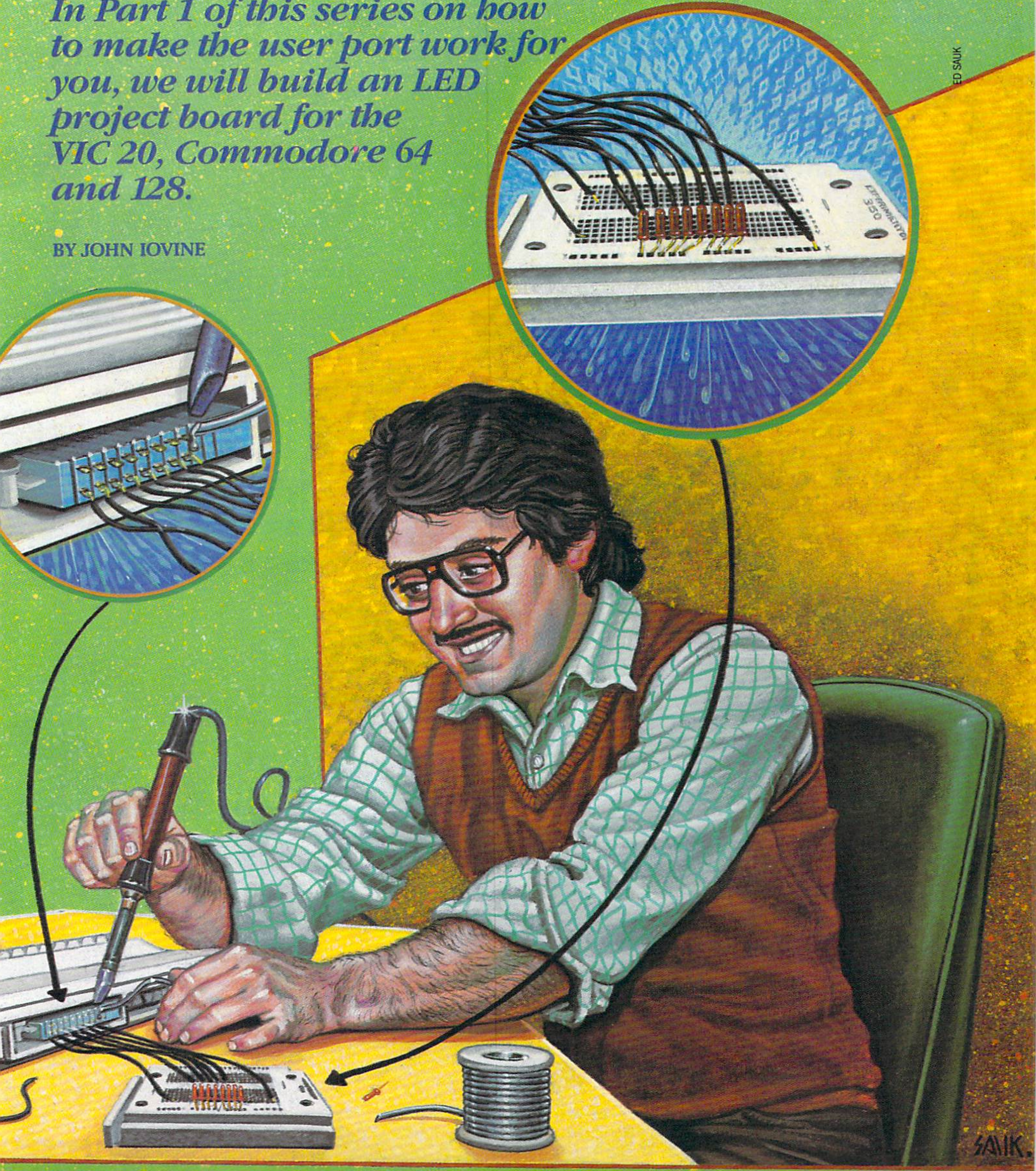
Binary means "based on two," as in two numbers, 0 and 1. Or like an electrical switch that has two values off



In Part 1 of this series on how to make the user port work for you, we will build an LED project board for the VIC 20, Commodore 64 and 128.

BY JOHN IOVINE

ED SAUK



(0) and on (1). In binary a digit is called a bit, which stands for binary digit. A byte is a digital expression containing eight bits. The microprocessor used in the computers we're working with are eight-bit microprocessors.

Since this article is on interfacing the user port, we will investigate the binary relationship to controlling various I/O functions. All the information, however, is applicable to controlling other chips in Commodore computers. A fuller understanding of binary number system can be acquired by reading any of the many books available on machine language.

As seen in Table 1, for each progression of the binary 1 to the left, the power of 2 is increased by 1. These are relevant numbers because each progression identifies a bit location and weight. Notice the correlation between the user port and bit weight table; we will be using this often.

Table 2 demonstrates binary counting 0 thru 20. This chart comes in handy later.

When a bit in the port is configured for input, the computer

Diagram A

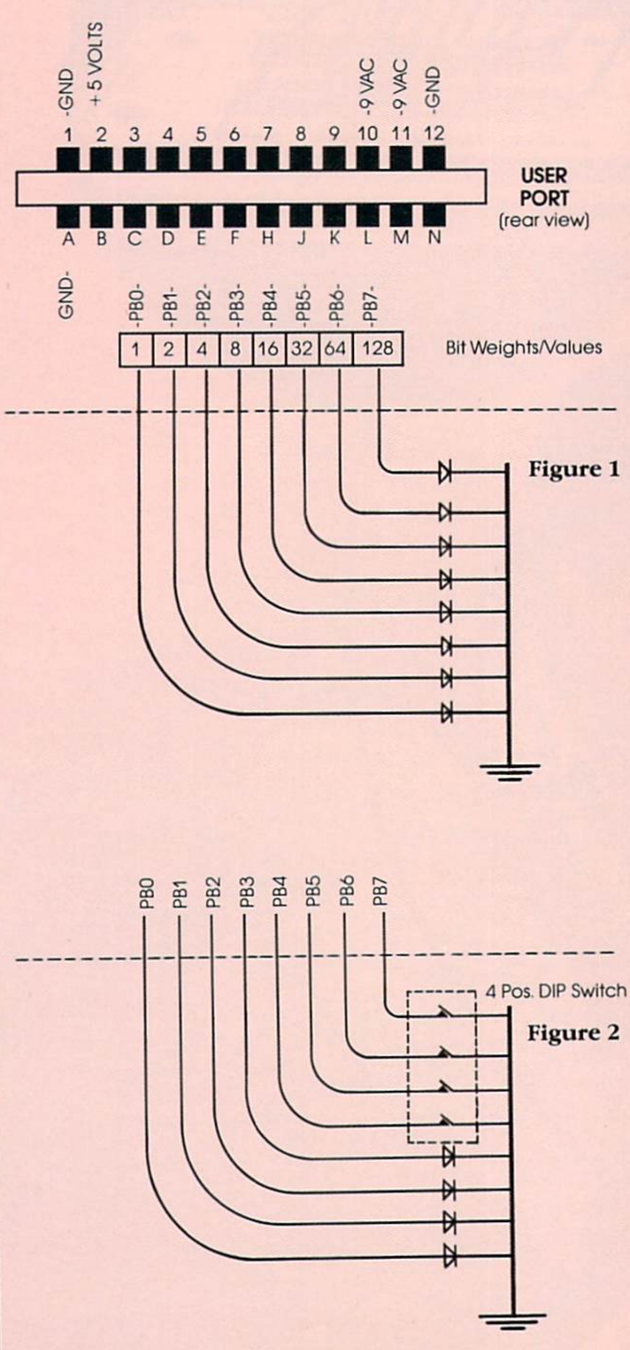


Table 1

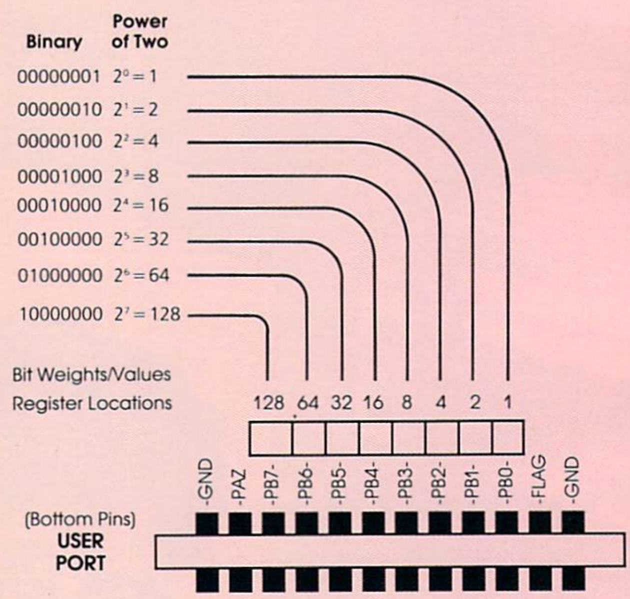


Table 2

Decimal	Binary	Decimal	Binary
0	=00000000	11	=00001011
1	=00000001	12	=00001100
2	=00000010	13	=00001101
3	=00000011	14	=00001110
4	=00000100	15	=00001111
5	=00000101	16	=00010000
6	=00000110	17	=00010001
7	=00000111	18	=00010010
8	=00001000	19	=00010011
9	=00001001	20	=00010100
10	=00001010	255	=11111111

uses electrical voltages present at the pin/bit to determine whether that bit is set (on) 1 or clear (off) 0. A binary 1 is equal to a voltage level between 2 and 5 volts. A binary 0 is equal to a voltage level between 0 and 0.8 volts. Voltages between 0.8 and 2 volts are undefined.

When a bit in the port is configured as an output, the computer will output five volts when a binary 1 is placed at the pin/bit location or zero volts when a binary 0 is placed at the pin/bit location.

DDR

The DDR (data direction register) is a programmable register on the 6526 chip that controls the direction of the bits in the port (input or output). A binary 1 placed at a bit location will turn that bit into an output bit. Conversely, a binary 0 will make that bit an input bit.

The DDR occupies one byte in memory. The location of the DDR for port B on the user port is 56579 for the 64 and 128. For the VIC 20, the DDR location is 37138.

We use our bit weights (Table 1) to output binary 1's at the corresponding pins to create output pins. Any pins that aren't programmed as outputs automatically have 0's placed at their bit location and are therefore configured as input pins (Table 2). POKE 56579,20 would turn PB2 and PB4 into output bits, as PB0,PB1,PB3,PB5,PB6 and PB7 automatically become input bits.

To see this more clearly, transfer the binary equivalent of 20 (Table 2) into the empty register location spaces on Table 1. The binary 1's are in PB2 and PB4 bit locations.

POKE 56579,3 makes PB0 and PB1 output bits transfer the binary equivalent of 3 into the location spaces. Doesn't the number 3 in binary place binary 1's at the location of PB0 and PB1? As you can see by poking this location with various bit weights, we can configure any pin in the port to be an input or output bit in any combination we might require. Any unused bits can be ignored.

To summarize, poking a binary 1 in the DDR corresponding to a bit turns that bit into an output bit. Conversely, poking a binary 0 will turn the bit into an input.

Peripheral Data Register

After we have configured our port with the DDR, we can start using it. The peripheral data register memory location is 56577. This is where we poke and peek to input or pull data off the pins.

The procedure is similar to the one described for the DDR. We'll go into greater detail after we build our demonstration circuit.

Input

Examine the diagram of the user port again (Diagram A). Beneath the user port are labels PB0, PB1, PB2,... PB7 corresponding to the pins on the user port. Under that is the corresponding bit weight for each pin.

Now let's configure all the bits on the port as inputs.

POKE 56579,0   DDR places binary 0's at all bit locations.

Now we apply +5 volts to pins PB2 and PB4. By applying the 5 volts to these pins we are inputting a binary 1 at each pin. If we then peek the port

PRINT PEEK(56577)   Peripheral Data Register

the number 20 would be returned. This is the added bit weights (4 + 16 = 20) of pins PB2 + PB4. Look at Table 2. Transfer the binary equivalent of the number 20 into the bit locations on Table 1 and it is the same. The binary 1's are in the same bit positions we inputted. If we applied +5 volts to just PB5, then peeked the port, the number 32 would be returned. This is true for all pin/bit combinations.

Output

Let's reconfigure our user port so that all the bits are now outputs.

POKE 56579,255   DDR places binary 1's at all bit locations.

Now poke the number 20 into the port.

POKE 56577,20   Peripheral Data Register

What do you think will happen? If you reasoned that +5 volts would appear on PB2 and PB4—you are right! By poking the number 20 into the port, we are essentially outputting a binary 1 at those two pins.

It is important to understand that the voltage being outputted is a signal voltage and has very little power. Therefore it cannot be used to run a device. But by adding a simple circuit described later, you can use that signal to control most any electrical appliance you'd like.

If you feel a little confused at this point, don't worry. It will all come together very quickly once you gain some practical experience by experimenting with the port. In order to do this, you will need to build the demonstration circuit.

Circuit Construction

Look at Figure 1. This is a simple circuit that doesn't require much soldering. Most of the soldering has been eliminated by using an experimentors plug-in breadboard. By using this board we can simply plug in our components and control lines. This board also facilitates changing the circuit by being able to pull out the components and replacing them with different ones.

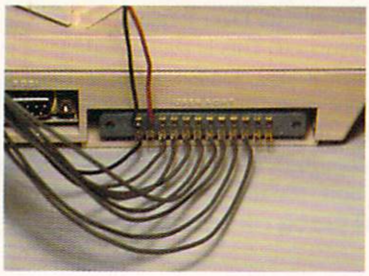
Parts List

Quantity	Item	Manufacturer	Part Number
8	Subminiature Red LED	Radio Shack	276-026B
1	Breadboard	Radio Shack	276-175 or 276-174
1	12/24 Card Connector	Mouser Electronics	568-50-24A-30*

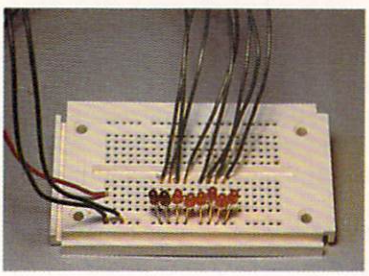
*Available @ $3.69 from Mouser Electronics
11433 Woodside Ave.
Santee, CA 92071
(619) 449-2222

All the parts required are available at your local Radio Shack Store, except for the 12/24 card connector. This is available from Mousser Electronics (see Parts List). You can utilize a 44-pin connector that is available at Radio Shack by cutting off one end, leaving 12 pins.

Take care in making your solder connection to the card connector. Notice that in addition to soldering wires to our data-port pins, we have also taken a lead off an additional ground pin and the +5 volt pin. These lines will be useful later on.



Do not substitute the LEDs. The LEDs used are micro-miniature; these were chosen because they don't require much power and can be lit directly from the current available at the port. When you have completed the soldering and wired the circuit as shown, we are ready to continue. Turn off your computer (if it is on) and plug the card connector into the user port. Power up your computer. All the LEDs should be dimly lit. If they are not, turn off the computer immediately because you've made a wiring error. Recheck your wiring and make sure the LEDs are in properly, facing the right polarity. The reason the LEDs are dimly lit is that although the computer configures all bits as inputs on power up, the 6526 chip pulls the bits to five volts through a 10K register. There is sufficient current to only dimly light the LEDs.



If everything checks out you're ready to continue. Enter

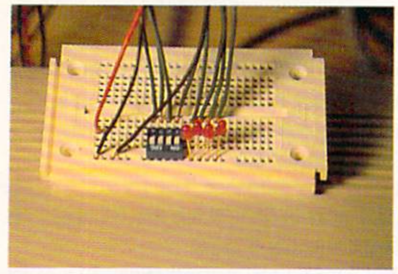
POKE 56579,255	Set up DDR. This turns all the bits to output bits.
POKE 56577,20	Peripheral data register lites LEDs connected to PB2 and PB4.
POKE 56577,0	Turns off LEDs.

In order to become familiar with the bit weights and their correlation to the pins, type in this simple program. Any number you input will light the LEDs corresponding to the bit weight.

```
10 POKE 56579,255
20 INPUT"[DOWN4] INPUT BIT WEIGHT ";BW
30 PRINT"[CLEAR,DOWN4]THE NUMBER"BW"
   IS BEING DISPLAYED IN BINARY ON
   YOUR INTERFACE"
50 POKE 56577,BW
60 GOTO 20
```

This second program will count in binary. To make it run faster or slower, change the value of T accordingly. To count to a value less than 255, change X accordingly.

```
10 POKE 56579,255
20 FOR X=0 TO 255
30 POKE 56577,X
40 FOR T=1 TO 255:NEXT T
50 NEXT X
60 GOTO 30
```



We now have some experience outputting binary 1's. Now let's reconfigure the port. First turn off the computer. Remove the four LEDs connected to PB4 thru PB7 and replace it with the four-position dip switch (see photo). Enter

POKE 56579,15	DDR set-up configures PB0 thru PB3 as outputs; PB4 through PB7 as inputs.

Turn all the switches on. Enter PRINT PEEK (56577) and a 0 will be returned. Turn off the switch connected to PB4 and reenter PRINT PEEK (56577) and now the number 16 has been returned.

You should know by now that the number 16 represents the bit weight for that pin. But the question remains, "Why is it that when you turn the switch off that the computer reads a binary 1 at the bit location?" The answer is the same as why the LEDs are dimly lit upon power up. When our switch is turned on, it is connected to and pulled to ground (0 volts). When the switch is turned off, the line is pulled up to +5 volts (binary 1) by the 6526 chip though a 10K OHM resistor. To become more comfortable with inputting, enter the following.

```
10 POKE 56579,15
20 D=PEEK(56577)
30 PRINT"[CLEAR,RIGHT7,DOWN7]
   "D "THIS IS THE BIT WEIGHT OF YOUR
   INTERFACE"
40 GOTO 20
```

By turning various switches on and off, the bit weights are displayed on the screen. Let's do something a little interesting. Press the RUN/STOP key and enter POKE 56577,5. The LEDs connected to PB0 and PB2 are lit. Enter RUN and notice that the bit weight of 5 is added to the display.

Now let's try something a little more interesting. Enter the following program. When bit 8 (dip switch connected to PB7) is turned on, the LEDs connected to PB0 to PB3 will start counting. No other bit has any effect. This is accomplished by masking all bits except bit 8 and reading its bit weight.

```
10 POKE 56579,15
20 FOR X=0 TO 15
```

Continued on pg. 106

Only NRI teaches you to service all computers as you build your own fully IBM-compatible microcomputer

NEW! Train with the newest Sanyo 880 Series Computer—it's fully IBM-compatible and runs almost twice as fast!

With computers firmly established in offices—and more and more new applications being developed for every facet of business—the demand for trained computer service technicians surges forward. The Department of Labor estimates that computer service jobs will actually *double* in the next ten years—a faster growth rate than for any other occupation.

Total systems training

No computer stands alone... it's part of a total system. And if you want to learn to service and repair computers, you have to understand computer *systems.* Only NRI includes a powerful computer system as part of your training, centered around the new, fully IBM-compatible Sanyo 880 Series computer.

As part of your training, you'll build this highly rated, 16-bit, IBM-compatible computer system. You'll assemble Sanyo's "intelligent" keyboard, install the power supply and disk drive and interface the high-resolution monitor. The 880 Computer has two operating speeds: standard IBM speed of 4.77 MHz and a remarkable turbo speed of 8 MHz. It's confidence-building, real-world experience that includes training in programming, circuit design and peripheral maintenance.

No experience necessary—NRI builds it in

Even if you've never had any previous training in electronics, you can succeed with NRI training. You'll start with the basics, then rapidly build on them to master such concepts as digital logic, microprocessor design, and computer memory. You'll build and test advanced electronic circuits using the exclusive NRI Discovery Lab®, professional digital multimeter, and logic probe. Like your computer, they're all yours to keep as part of your training. You even get some of the most popular software, including WordStar, CalcStar, GW Basic and MS DOS.



Your NRI total systems training includes all of this: NRI Discovery Lab® to design and modify circuits • Four-function digital multimeter with walk-you-through instruction on audio tape • Digital logic probe for visual examination of computer circuits • Sanyo 880 Series Computer with "intelligent" keyboard and 360K double-density, double-sided disk drive • High resolution monochrome monitor • 8K ROM, 256K RAM • Bundled software including GW BASIC, MS DOS, WordStar, CalcStar • Reference manuals, schematics, and bite-sized lessons.



NRI is the only technical school that trains you on a total computer system. You'll install and check keyboard, power supply, disk drive, and monitor, following step-by-step directions.

Send for 100-page free catalog

Send the coupon today for NRI's 100-page, full-color catalog, with all the facts about at-home computer training. Read detailed descriptions of each lesson, each experiment you perform. See each piece of hands-on equipment you'll work with and keep. And check out NRI training in other high-tech fields such as Robotics, Data Communications, TV/Audio/ Video Servicing, and more.

SEND COUPON TODAY FOR FREE NRI CATALOG!

NRI SCHOOLS

McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, NW, Washington, DC 20016

We'll give you tomorrow.

For Career courses approved under GI Bill
☐ check for details.

☑ CHECK ONE FREE CATALOG ONLY

- ☐ **Computer Electronics**
- ☐ TV/Audio/Video Servicing
- ☐ Satellite Electronics
- ☐ Robotics & Industrial Control
- ☐ Data Communications
- ☐ Industrial Electronics
- ☐ Communication Electronics
- ☐ Electronic Design Technology
- ☐ Telephone Servicing
- ☐ Digital Electronics Servicing
- ☐ Basic Electronics
- ☐ Electricians
- ☐ Appliance Servicing
- ☐ Small Engine Repair
- ☐ Air Conditioning, Heating, & Refrigeration
- ☐ Locksmithing & Electronic Security
- ☐ Building Construction
- ☐ Automotive Servicing
- ☐ Photography
- ☐ Bookkeeping & Accounting

Name (Please print) ____________ Age ____

Street ____________

City/State/Zip ____________

Accredited by the National Home Study Council

#235-047

BY ELIZABETH DEAL

Universal Program Lister, *Part 1*



In Part 1 of this two-part series, the history of Commodore BASIC reveals why BASIC programs on one Commodore computer may not list correctly on another. Part 2 features a universal listing routine to bridge the gap between computers.

Tokens affect the listing and entering of a program into a computer different from the one on which it was written.

Commodore computers have a built-in table of BASIC keywords. The table is in numeric order, and all numbers are higher than decimal 127. BASIC programs are tokenized, which means that a word such as INPUT is stored not as five letters, but as a token that takes up only one character or byte. For example, INPUT's token number is 133 or $85 in hexadecimal notation.

Table 1 groups all keywords by their numeric token values. The values are in hexadecimal notation because hex numbers take up only two columns.

Why is all this important? It is important because tokens affect the listing and entering of a program into a computer different from the one on which it was written. What does a listing routine normally do? It sees a number such as $85, goes to the table of keywords, subtracts $80, and ends up with 5. The last letter of every keyword in the table is shifted, so every time there is a shifted letter, it indicates an end of a keyword. Count the shifted letters and you get a keyword number. The listing routine then prints all the letters of the fifth keyword (counting from zero) on the list, which is INPUT.

This is useful to see how programs are "crunched" when a program is entered into a computer. When you code 10 INPUT Z and push RETURN, the screen editor and the BASIC editor go to work. The editor sees a number 10 and takes care of it. Then it sees letters I-N-P-U- T. To see if it is a keyword, the editor begins scanning the keyword table. Every time it stumbles on a shifted letter it increases the count. Five hops later it lands on a matching set of letters, our INPUT. It assigns a value of the counter, now 133 or $85 in hex and puts that in memory.

This takes a bit of time. The longer the keyword table, the more time it takes. On the Commodore 128 the time is quite noticeable. What if the list is exhausted? It can't be a keyword. The computer leaves the BASIC text intact in a form of an ASCII string. So in INPUT Z, Z is regarded as a string. Likewise, if you coded PRINT"some string," PRINT gets tokenized and what's in quotes is not. Finally, GOSUB 5000 tokenizes the GOSUB part and leaves 5000 in ASCII form.

Suppose you coded on the Commodore 64 100 INPUTA$:IFA$="N" THEN DCLOSE and planned to run this code on the 128. What a mess! You'd load this program into the 128 and DCLOSE would do nothing except tell you of a syntax error. Why? Because the 64 has no DCLOSE (close all disk files) keyword. So it left the letters D-C-L-O-S-E as an ASCII string, unchanged. A transfer to the 128 does not change the string into a token corresponding to DCLOSE, resulting in the error message.

How to make it work in another computer? Simple—locate the line in trouble and push RETURN. The instant you push RETURN, line 100 will be tokenized in the host computer and DCLOSE becomes a valid token $FE-0F. If you need more changes to be made, mechanical means are necessary.

Paths of Keyword Development

Now let's make some sense out of all this. Once, a long time ago, Commodore built the PET computer. It had a vocabulary of about 74 words. Of course, since then things have changed and computers have many more words. Look at the keyword chart. It shows boxes containing lists of keywords in all CBM computers to date. The original, BASIC1 PET, knew only the left-hand side column of BASIC keywords.

The PET, VIC 20, 64, B128/256 Keyword Path (upper half)

Life was simple in Commodore land for awhile. Pure BASIC programs could move from one machine to another without trouble. But when BASIC 4 PET's appeared, things changed a bit. A user with an original or BASIC 2 PET could no longer list all the programs that were written. A program written on a BASIC 4 PET had keywords the earlier PET's could not cope with. NEXT would list instead of BASIC 4's DCLOSE.

Soon programs like Jim Butterfield's Lister appeared on the scene to help users list the "foreign" program. Once the user got a sensible listing, he or she was usually able to convert the program to run on the older PETs.

To this day, the vocabulary of the PET, Commodore 64 and the VIC 20 is that of BASIC 2 PETs, which includes keywords from the original PET (left-hand column) and the keyword GO. BASIC 4 PET computers have additional keywords, with numbers sequentially higher than GO (top half of the chart), keywords numbered $CC-DA, or words CONCAT to DIRECTORY.

The famous Protecto-special B128 machine has the richest language in that group. It contains keywords numbered $DB to E8, or DCLEAR to INSTR. The B-machine keywords are unknown in the BASIC 4 PET's and cause similar problems to the DCLOSE example.

Still, the keyword boxes shown on top of the picture show that the development was linear, that is, new keywords were added at the end of the existing list. This design permits people to move programs from one computer to another with little difficulty. Usually, just a little bit of conversion is needed to make programs work in other computers once you have an accurate listing.

The 64 Expander Path (from PET to upper right-hand side)

One exception to the linear expansion was the addition of Commodore's Super-Expander cartridge for the 64. The next available token number would have been $E9, but instead Commodore created double tokens. See the $FE list on the right-hand side of the chart in the upper portion. These keywords cause problems in the PET's and the B-machine.

Listing a program containing the FE keywords on any of the PET's relatives produces garbage: The $FE character prints as a graphic quarter-square, followed by cursor down, a space, a color change in the case of the 64 or VIC 20 and so on. It all depends on that number after $FE.

The PET line of computers was not equipped to handle double tokens. These numbers basically become control characters and cause an unreadable mess. Of course, an expander program cannot run in any non-expandable computer.

The Plus/4 and 128 Keyword Path (lower half)

In 1984 the Plus/4 entered the scene. It has a huge vocabulary to support all the features the 64 had, but lacks the

To this day, the vocabulary of the PET, Commodore 64 and VIC 20 is that of BASIC 2 PETs.

keywords for sound and graphics, as well as disk commands similar to BASIC 4 PETs.

Now you'd think that the new graphic and sound keywords would go at the end of the list (after $E9). You'd also think that keywords common with BASIC 4 PET's and the B-machines would have the same token numbers. Neither is true. We now enter a new path, a lateral diversion (see lower half).

In the $CC column, there are two keywords: RGR and RCLR. In the PET's path we saw CONCAT and DOPEN. So if you load a BASIC 4 PET program that opens a file using the DOPEN command into the Plus/4 computer, it lists as RCLR; same token number. It will not open files and most likely will return a syntax error since RCLR needs parameters different from those of DOPEN.

The original program meant to open a file. Can a user of a Plus/ 4 or 128 computer DOPEN a file using a BASIC 4 PET program? You bet. Just edit RCLR to DOPEN, push RETURN, and you're in business. From now on this is a genuine Plus/4 program. However, this sort of a conversion in a long program is a beast. You have to have a conversion chart in hand, and you'll have to edit lots of lines.

In column $CE our paths diverge laterally once again. The Plus/4 takes the upper route via RLUM. The 128 takes the lower path via a table of several double tokens. $CE tokens are most complicated to deal with. In the entire CBM line we see three variations in this group: PET 4, Plus/4 and 128. As was the case with the expander programs $CE double tokens of the 128 list funny in other computers. Once again control characters can kill you.

By the time we reach the $CF column, the Plus/4 and 128 paths merge into a large group of the common keywords JOY to WHILE, or $CF to FD. So here, at least, Plus/4 and 128 speak a common language. But the language is incompatible with the PET's type of keywords in several respects. They do not list correctly, of course, and the host computer can even crash. If a program containing keyword number $E9 (SCALE) is loaded, BASIC 4 PET and the B-machine go bye-bye. Keywords higher on the list than $E9 do not cause a crash.

And finally, the Plus/4 repertoire may include on a cartridge custom keywords with number $FE in the first position. If that happens, nothing can be done unless everybody has the cartridge. In any case, Plus/4 ends at $FD.

The 128 has a long list of keywords in the $FE double-token group (see the bottom right-hand side column). You'd think that there might be some correspondence to the 64 expander cartridge keywords in the same $FE column, but there's not. The 64's KEY will list as BANK on the 128 (or it won't list at all). The 128's RSPPOS ($CE-05) corresponds to the 64's $FE-9E.

It is surprising how well many programs move from one computer to another, especially because the potential for confusion is so great. Here are just some examples of what can happen.

A BASIC 4 PET program containing COLLECT ($D1) lists as DEC in the Plus/4. Conversely, DEC in the 128 lists as COLLECT in the BASIC 4 PET or B-machine.

A Plus/4 program containing COLLECT ($F3) lists as garbage and will not COLLECT anything in the BASIC 4 PET or B-machine. For that we need keyword number $D1.

The B-machine's INSTR ($E8), a string function, lists as SCNCLR (screen clear) in the Plus/4. Chances are that a B-program loaded into a Plus/4 will actually clear the screen and give a syntax error on the parameters of the INSTR command.

Plus/4's INSTR is keyword $D4. It will list as APPEND in the BASIC 4 PET and B-machine.

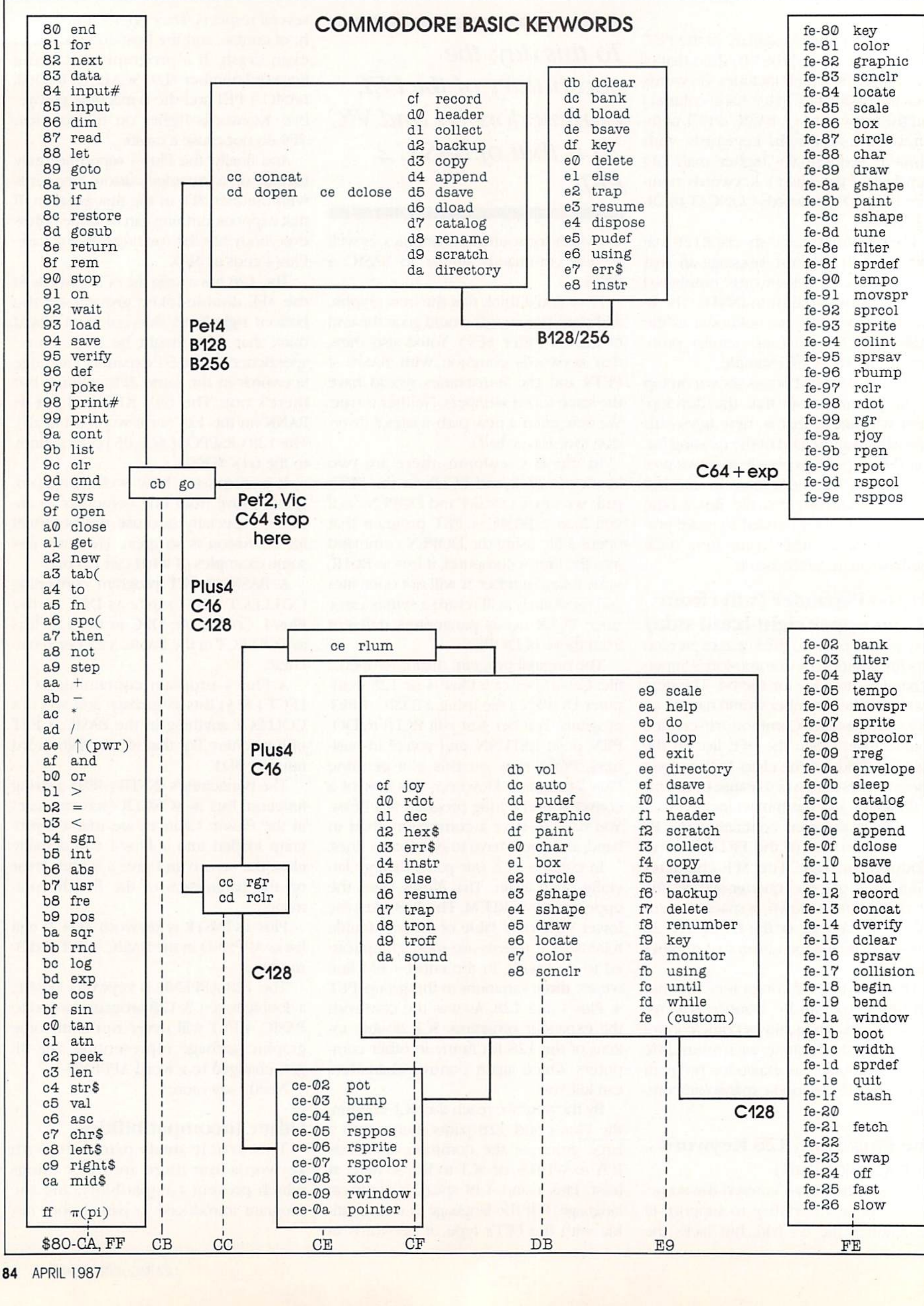
The 128 APPEND is keyword $FE-0E, a double token. A 128 program moved to BASIC 4 PET will never run until some graphic garbage representing $FE-0E gets changed to a word APPEND.

Need I say more?

Other Incompatibilities

This article deals primarily with keywords. But there are other things which prevent compatibility, and the program introduced in part 2 does not

COMMODORE BASIC KEYWORDS



Cross development has to be done with care and attention. If you plan to move programs from one computer to another, it pays to learn a little about the other computer.

deal with them. These include things like sprites in a computer that has none, SID sound in a computer that has no SID chip, high-resolution graphics in a computer that has no bit-mapped display, and wrong POKE or PEEK addresses for various machines (screen start being the most common one).

Less obvious examples include color assignments in PRINT statements in the 64 that may do nothing on the PET or may make a mess. For instance, "black" typed in as a graphic on the 64 will cause BASIC 4 PET to [shift]-ESC the quote mode!

There are also graphic problems like the 22-40-80 column screen widths. The PET's backslash shows up as an english-pound on all non- PET computers. Some character sets are slightly different. And watch strings in PRINT statements: Color assignments are different between the 64, 128 and Plus/4. The 64 POKE color statements need a value one higher when moved into the Plus/4 COLOR command.

Delay loops function differently. The VIC 20 is the quickest machine of the whole bunch; the Plus/4 and 128 are slow. B-machines are so fast that you may need to adjust the timing to suit a program.

There are differences in reserved variables. ST exists in all computers. TI exists in all except the B-machine. TI$ is six digits long in all computers but the B-machine, where it is seven digits long (time is taken from the chip, includes ten-of-seconds, which creates a lovely complication in all programs that depend on the jiffy clock). The BASIC 2 PET, 64 and VIC 20 do not know about BASIC 4 PET disk status variables DS and DS$ (you'll need to record as shown in the disk manual—open channel 15 and input the status variables, else DS is always zero and you'll never catch disk errors!).

DS or DS$ moved from a 64 or PET into a BASIC 4-type computer will return a syntax error. You'll need to change the names of those variables for a program to run. None of them know about B-machine's ER, EL, ERR$ reserved variables for error trapping and so on and so forth. Cross development has to be done with care and attention to these small items. If you plan to move programs from one computer to another, it pays to learn a little about the other computer.

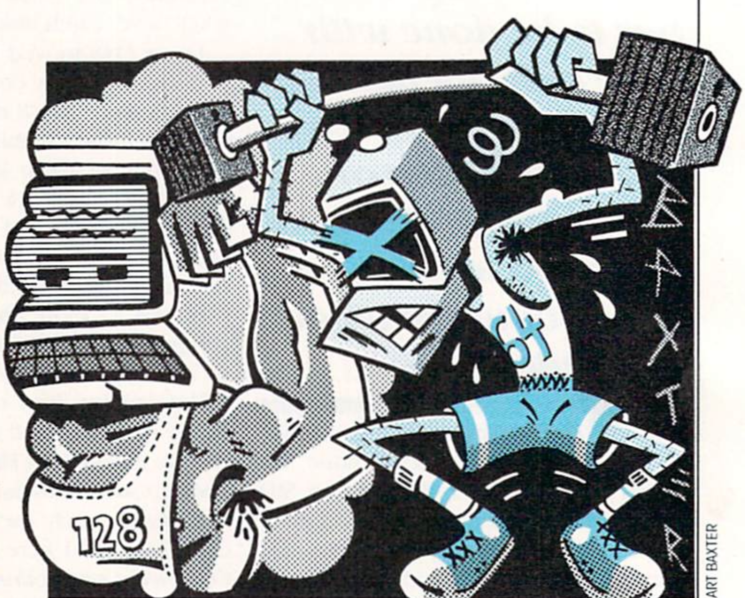
But, you say, how can I possibly have all the manuals to all the other computers? No need—Karl Hildon, the editor of *The Transactor* has published a book that includes precisely the set of cross- references we need here. The book is *The Complete Commodore Inner Space Anthology* and retails for $14.95. The address in the United States is 277 Linwood Avenue, Buffalo, NY 14209-9990. In Canada, the address is 500 Steeles Avenue, Milton, Ontario L9T3P7 Canada. The Canadian phone number is 416-876-474.

What we need is a listing program—one that can list a foreign program so you can study the programmer's intention. Once you know that the 128 programmer meant to APPEND more information to a file on disk, you can edit the silly looking version of the program on the BASIC 4 PET's screen to actually say APPEND.

The same procedure goes for all the other tokens or computer combinations. A good listing program is definitely the key. It is invaluable to user groups who swap programs between computers, as well as to people who develop programs on one computer to run on another. Part 2 of Universal Program Lister includes this program. C

BY IAN ADAM

Upgrading Your Commodore 64 to a Commodore 128



ART BAXTER

If you are considering upgrading your 64 to a 128, see just how easy the transition can be.

If you're considering upgrading from a 64 to a 128, you may be wondering just how easy the transition will be. Well, take it from me—the 128 is compatible with the 64 in every way, from existing software to peripherals and programming.

Bringing It Home

The good news is that the 128 is completely plug-compatible with your 64. Just unplug the disk drive, monitor, printer or other accessories from the 64 and plug them straight into the 128. Everything works with no modifications or special cables. Of course, you won't be able to take advantage of the added features of the new disk drive and monitor, but at least you'll be computing with the 128 with no further expense. The same is true of VIC 20 peripherals, although the software is generally not transferable.

As for your program library on the 64, it will work on the 128 as long as you are in 64 mode. With a cartridge, this is automatic— just plug it in and turn on the 128. Without a cartridge, the computer will be in 128 mode when you turn it on. Just type GO 64, answer Y to the prompt ARE YOU SURE?, and up comes the familiar blue- on-blue screen of the 64.

If you have a cartridge like Epyx's Fast Load, it will automatically perform these steps for you, putting the computer into 64 mode and setting up its usual features. You can then proceed to load your programs from disk or tape in the usual fashion. All of the programs I tried worked perfectly.

It is not so much a question of whether the 128 is compatible, but that there is an entire 64 being held captive inside the somewhat larger case of the 128!

While it's nice to know that your accessories and software will work unmodified on the 128, there is also the question of how to take advantage of all those extra features, the extra memory, enlarged BASIC, and CP/M capabilities. After all, since you have so much more power in the new machine, there's no point using it in 64 mode all the time.

Upgrading BASIC

Let's take a look at how your BASIC programs can be promoted from the 64 or VIC 20 to the 128. Turn on the computer with no cartridge, but this time stay in 128 mode. Again, loading the program is no problem, as your existing peripherals will do the job. With the 1541 disk drive, the usual LOAD "program",8 will work fine. It is much easier, however, to press F2, which prints DLOAD to the screen. Then just enter the program name and press RETURN to load it.

The two versions of BASIC are upward-compatible. What this means is that all of the new enhancements are additions to the commands available on the 64. As a result, all of the commands in the 64 work the same in the 128. The same is not true in reverse, however. Programs written on the 128, if they use the new commands for graphics, sound and utilities, cannot be executed by the 64. If you try this, you will simply get syntax errors.

This doesn't mean, however, that programs from the 64 will work unmodified in 128 mode. The main culprit is that ubiquitous command POKE and its mild-mannered cousin PEEK. Because the 64 lacks those commands, any program wanting to use graphics or sound must do its thing by poking appropriate values into memory or into the control registers of the video and audio chips. In 128 mode, the memory configuration is different—running a program written for the 64 would not necessarily achieve the expected results. Accordingly, promoting a BASIC program from the 64 will consist of four steps: load the program into 128 mode; scan the program for pokes and peeks, and replace these with the new 128 commands; use your creativity to improve the program to take advantage of the added capabilities; and save the upgraded program, preferably on a 1571.

The first step is easy. But let's take a look at what's involved in the conversions in the second step. We'll also look at a specific program to see how the new graphics work.

BASIC Conversions

The meaning of a POKE command can be determined by examining the range it pokes into. For example, POKE

53280,1 pokes into the video chip range. More precisely, it causes the screen border to turn white. Here are some of the ranges to look out for.

Range	Effect	Replace with
53248- 53279	video chip	New graphics commands COLOR SPRITE, etc.
54272- 54300	sound chip	Sound commands PLAY, ENVELOPE, etc.
0 to 1023	operating system	Check a memory map
1024 to 2023	screen memory	Ok, or use PRINT

Let's look briefly at each of these ranges. The table lists the main substitutions you will have to make.

Video Chip

The video chip is probably the area where there has been the greatest improvement in commands. Graphics 'programs were previously a mass of pokes. Now graphics can be created with some simple commands. These accomplish many effects that could only be achieved with machine-language programming previously.

The first 17 registers control the x and y coordinates of sprites. Replace pokes to these values with the new MOVSPR x,y command. This command can also be used to give sprites independent movement. Because sprites in the 64 can move off the screen, the new coordinates will always be less than the old poke numbers. Note too that one SPRITE command can replace up to six pokes.

The numbers that you use to set screen colors are one higher than before and now correspond to the top row of the keyboard. For example, red, which is on the 3 key, always required poking a 2 with the 64, but now is achieved with a COLOR x,3.

VIDEO CHIP:		
sprite coordinates:	POKE 53248,100 POKE 53249,125	MOVSPR 1,76,75
light pen x:	PEEK(53267)	PEN(0)
light pen y:	PEEK(53268)	PEN(1)
turn sprite 1 on:	POKE 53269,1	
sprite 1 in red:	POKE 53287,2	
in front of text:	POKE 53275,0	SPRITE 1,1,3,,1
expand x direction:	POKE 53277,1	
don't expand y:	POKE 53277,0	
one color mode:	POKE 53276,0	
set border color:	POKE 53280,n	COLOR 4,n + 1
background color:	POKE 53281,n	COLOR 0,n + 1
SOUND CHIP:		
maximum volume:	POKE 54296,15	VOL 15
voice 1 ADSR:	POKE 54277,17 POKE 54278,135	ENVELOPE n,1,1,8,7
voice 1 middle C:	POKE 54272,195: POKE 54273,16: POKE 54276,17 : FOR I = 1 TO 100: NEXT: POKE 54276, 16	PLAY "C"
sound effects:	about 20 POKEs:	SOUND 3,256,500,,,
read paddles:	PEEK(54297/8)	POT(1)/POT(2)
OTHER:		
read joysticks:	PEEK(56320) PEEK(56321)	JOY(1) JOY(2)

Sound Chip

The new commands allow music to be played in a very natural fashion. You can throw out your long series of pokes and tables full of frequencies, and replace them with a simple PLAY "ABC...". While you can set the attack, decay, sustain, release and waveform if you want, there are ten preset instruments that can be used without any further effort.

Note a small error in the documentation for the PLAY command. You use Un to set the volume, and the system guide says n can be 0 to 15. The range is 0-9.

Screen Memory

Screen memory is in the same place in the 128, so pokes into the range 1024 to 2023 can remain. However, with improved screen formatting, you should replace these with PRINT statements.

Operating System

Any poke into the area below 1024 is likely to affect the operating system. You will need a memory map and the knowledge to use it in order to deal with these commands.

Running Speed

Some programs may be affected by variations in run speed, particularly games that depend upon timing. Using the three-line program shown to time the computers, I found very little variation. The 128 in 64 mode is about 3% faster than the 64. In 128 mode, it turns out to be 5% slower, due no doubt to the extra work associated with the larger memory and fancier screen controls. The 128 also has a fast mode, which is just over twice as fast as slow mode.

More Tips

I hope this information will give you some assistance in promoting your BASIC programs from the 64 onto the 128. Once the conversion is made, you can use the new commands to enhance the programs greatly. If there is enough interest, I'll try to cover some of these enhancements in the future. In the meantime, here are a couple of hints to give you a preview of the additional features you can get with the 128's own peripherals.

You will need the 1902 monitor to get full access to all graphics modes. However, you can get a sneak peek at 80 columns with your 1701 or 1702 monitor and a simple cable. You will need a 9-pin male subminiature connector and a commonly-available RCA cable. Connect the cable's ground wire to pin 1 and the center wire to pin 7. Plug the 9-pin connector into the computer port marked RGBI and the RCA plug into the video socket on the front of the monitor. Flick the signal-select switch on the back of the monitor and you're looking at the 80-column screen (monochrome only). Press ESC then X to start using the screen. The resolution isn't great, but at least you'll know it works.

Contrary to some rumors, you can load CP/M from the 1541 disk drive. Simply insert the CP/M system disk, type BOOT, and hit RETURN. The problem will arise in transferring CP/M programs into a format the 1541 can understand. If a friend who's an electronics engineer can do that for you, great. Otherwise, you will need the new 1571 drive to read standard CP/M disks.

That's all for now. The accompanying program is just a dem-

onstration, but it shows how easily a graphics demonstration can be written in BASIC. The same thing on the 64 would require literally hundreds of pokes. Try modifying the starting values of X1, X2, and DC to get different effects. ■

Before typing these programs, read "How to Enter Programs," and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Demonstration Program

```
140 X1=90: Y1=20: DX=8: DY=5
    :REM CHANGE...'FCRI
150 X2=170: Y2=80: EX=2: EY=-5
    :REM THESE'GYAK
160 XL=0: YL=0: XM=319: YM=199
    :REM DON'T CHANGE THIS!!!'FNQO
170 COLOR 1,8: GRAPHIC 1,1'CHIF
180 :'ABHD
190 DO: DO'CBQG
200 FOR I=1 TO 10'DECX
210 :'ABHW
220 X1=X1+DX: Y1=Y1+DY'ENFE
230 X2=X2+EX: Y2=Y2+EY'ENLF
240 :'ABHA
250 IF X1>XM OR X1<XL THEN DX=-DX
    : X1=X1+DX'JTVM
260 IF Y1>YM OR Y1<YL THEN DY=-DY
    : Y1=Y1+DY'JTFN
270 IF X2>XM OR X2<XL THEN EX=-EX
    : X2=X2+EX'JTDO
280 IF Y2>YM OR Y2<YL THEN EY=-EY
    : Y2=Y2+EY'JTMP
290 :'ABHF
300 DRAW 1,X1,Y1 TO X2,Y2'CMBB
310 DRAW 0,X1(I),Y1(I) TO X2(I),
    Y2(I)'CYEE
320 :'ABHY
330 X1(I)=X1: Y1(I)=Y1'CPWE
340 X2(I)=X2: Y2(I)=Y2'CPBF
350 :'ABHC
360 NEXT'BAED
370 :'ABHE
400 CHAR ,18,23, "PRESS A KEY TO
    PAUSE"'BHDF
410 CHAR ,18,24, "PRESS Q TO
    QUIT"'BHXF
420 :'ABHA
430 GET A$'BCGC
440 LOOP WHILE A$=""'DCAF
450 GET KEY A$'CCEE
460 LOOP UNTIL A$="Q"'DCFH
470 :'ABHF
480 GRAPHIC 0: END'CCYH
490 :'ABHH
```

END

Timing Program

```
10 TI$="000000"'BDCY
20 FOR I=1 TO 100:A=I^2:NEXT'GKSD
30 PRINT TI'BCDA
```

END

Continued from pg. 24

into mercenaries, armed rabble, the needy, satanists, survivalists, local gangs, or foreign soldiers. Some encounters enable you to recruit gang members, while others spark foot combat. In foot battles you merely watch a combat report in the text window until one side is wiped out. It's fast, but offers no control over the action.

Run across another road gang and you get to choose from three combat modes: abstract, quick or detailed tactical combat. The first is like foot combat—speedy but dangerous. Lack of control over your party means you usually lose more gang members and cars than with the other combat modes. In these you decide how to deploy your gang by placing them on top of and inside your vehicles. Quick combat lets you select your cars' targets and make a few other tactical decisions.

A six- to eight-line message fills the screen to describe each attack, scrolling up until a victor is declared. The fully detailed tactical option—in which you direct your cars individually in a battle that can last from 15 to 30 minutes—is the heart of this game. Each battle occurs on a map that reflects the terrain in which the encounter takes place: city streets, forest, or an interstate highway are some of the possibilities. All cars are illustrated with your choice of icons seen from a side or aerial view.

First you give orders to move each car and change speed, then you pick specific targets for each vehicle. The results are animated, but far slower than in *AutoDuel*. This is a strategy and tactics battle, not a contest of coordination and reflexes. My favorite part of detailed combat is the transfer-and-board option. Here you can have your gang members jump onto adjacent cars to attack the crew and try to capture it, the ways pirates seized ships on the high seas. (It is not animated, however). Success requires using each vehicle's strong points. For example, a sports car is fast, while a bus packs so many men and weapons that it can be used like artillery in a war game.

You begin with six cars and must fight these lengthy tactical battles in order to expand your fleet. For each victory, you can obtain another car, up to 15. (I know people who finished the game with only six vehicles, relying mainly on quick combat.)

To find the scientists, you will roam from city to city and use the "people" command. (After you've found most of them and returned them to headquarters, you'll get a homing device that speeds up locating the others.) Eventually three specialized crew-types called Cronies will turn up: doctors, drill sergeants and politicians. Each has special abilities that improve your gang's performance. At times you must forego the quest for the scientists to seek fuel, food or medicine.

And occasionally you'll be prompted to make decisions, such as the time I visited Las Vegas and had to decide whether to let my crew go gambling or not. In these situations you'll discover that your decisions may affect the gang in an unexpected manner. Once half my crew deserted me and joined a religious cult!

There are no sound effects and the animation is slow. But this game's soft spot is really disk access. It's fast enough, but occurs almost every time you punch a key. Certain elements of the game, such as your starting place and the location of the scientists, are randomized for each new game, which increases the replay value. Even after the quest is over, you can return to the streets and play a detailed combat scene as a mini-game in itself.

So if you want arcade action and the opportunity to design your own cars, *AutoDuel* is the way to go. *Roadwar 2000* offers more for the player who prefers to take his time before making a move, who prefers to direct the actions of a fleet of cars in a war game-style environment. Either way you're likely to take a wrong turn somewhere up the road, so here are a few tips.

In *AutoDuel* spend all your character points on Driving and Marksmanship, since you can always boost your Salvage skill by paying for lessons in a garage. A safe way to increase these skills is to leave town just long enough to kill two or three outlaw cars, then return to the same town—don't try to make it all the way to another town early in the game. Your "save disk" is not copy-protected, so duplicate it. This way your character won't be lost if he gets killed and you don't have a clone.

In *Roadwar 2000,* get a big vehicle like a tractor trailer to carry lots of food. During detailed combat, turn your bus sideways or drive between the enemy so you can fire at the enemy from either side. Visit Mountain View early to find Cronies.

BY HENRIK MARKARIAN

Inter Melody
for the Commodore 64

The Commodore 64 possesses one of the best sound chips found in a home computer. However, using this sound chip (SID) is not easy, and at times very confusing. Inter Melody simplifies entering musical notes such as C,F,A# and also livens up your BASIC or machine-language programs.

In order to play a note, one must set the ADSR envelope, choose a waveform, and last but not least, determine the proper frequency for that note. The last of these tasks is the most cumbersome of all. Having to look at a table and picking out numbers for a desired note can be a difficult and a lengthy process.

By using Inter Melody you can enter notes as you would read them. For example, you enter a C note simply as C, or a G# note as G#. There is no need to look at the frequency tables any longer. However, Inter Melody does more than simplify editing music. The music that you enter will be played during system interrupts, thus the micro processor is free to do other chores. This means that a BASIC program or a machine-language program can run while the music is playing in the background. This feature can spice up almost any game that you can think of. Imagine playing the song from a popular movie while the game imitates the movie itself, or to speed up the song in the background once a higher level is reached in a game. Your imagination is the limit.

In order to use Inter Melody properly, follow these directions. First type in Inter Melody Code and save it. This is the machine- language part of the program, and it has to be in memory before any music can be played.

Next type in Inter Melody and save it. This is the BASIC part of the program. To run the program, first load and run Inter Melody Code. When you see the READY display, load and run Inter Melody. Since Inter Melody plays the music during interrupts, only two of the SID's voices are used. This is done so as not to slow down the operation of the microprocessor. Therefore it is up to the user to decide which of the two voices will be used.

One advantage of having only two of the voices play lies in the fact that one voice will be left free. For example, if you have a BASIC program that uses one of the voices for sound effects, you can tell Inter Melody to use the other two voices to play a song. This way, while your program is running, the music will be playing in the background. Once you choose which two voices, 1&2 or 1&3 or 2&3 will be used, the program will start reading the music data that you have entered.

The music must be entered via DATA statements starting at line 1000. A typical line might look like this: 1000 DATA C,C#,D,D#,E,F,F#,G,G#,A,A#,B,+,C,2,-,C,RES,END The following table explains all the symbols that can be used with Inter Melody.

Format	Meaning
C	Note C
C#	Note C#
D	Note D
D#	Note D#
E	Note E
F	Note F
F#	Note F#
G	Note G
G#	Note G#
A	Note A
A#	Note A#
B	Note B
+	Move up an octave (initial octave = 4th)
-	Move down an octave
RES	Rest (no sound)
END	End of data for one voice

You may also specify a duration for each note. Once you enter the note, follow it with a number, separate from the note by a comma. This number for the duration can range from 2 to 1024. However, if no duration is specified, a default duration of one is assumed. In any case, make sure that the number of beats (number of notes plus the number of specified durations) for one voice is equal to the number of beats for the other voice. Otherwise Inter Melody will respond with an error. This is because Inter Melody loops around once the music data is finished and if the number of beats per voice is not the same, the sound will be disoriented.

Once you finish entering the music data for one voice, you must signal Inter Melody with an END in the data statements. Then proceed with the music data for the other voice, and similarly finish it off with another END statement. Once all the data has been processed, you no longer need Inter Melody to play the music. All you do is type SYS36818 to start the music, or to stop the music if it is already playing. The first thing that you will notice is that the cursor does not disappear. Move around the screen or list the program—the music never stops.

The ADSR envelope and the waveform for each voice can be changed by simply changing the corresponding variable in lines 140 and 150. Hence, AD(1) refers to the attack decay rate for the first voice and AD(2) refers to the attack decay rate for the second voice. Similarly SR(1) and SR(2), refer to the sustain release rate for the first and second voices, respectively. The HP(1) and HP(2) and the LP(1) and LP(2) are used to set the pulse width if the pulse waveform is selected for any one of the two voices. The variables WA and WB refer to the waveforms chosen for the first and second voices, and finally TEMPO represents the time, in 60th of a second, that each note is maintained.

For example, if TEMPO equals 240, then each note is maintained for four seconds (240/60). It is best to set these parameters after the data has been processed. At that point you may change the parameters and then respond with a YES, once the program asks if the data has been processed. Continue this until you like a particular setting, and then save the program with the newly added DATA statements.

The ADSR envelope and the volume can be controlled by poking the desired value into the proper SID registers, thus enabling you to control these parameters from within a program. However, in order to change the waveform you must use Inter Melody. The tempo can also be changed from within a program, via the following poke.

POKE762,(0-255)

The highest address used by BASIC is lowered in order to store one part of the program and to set aside a 4K area for the music data from \$9000 to \$9FFF. Thus there is room for a total of 1,024 notes. The second part of the program, the actual wedge, resides in page two from \$02A7 to \$02FF. Therefore there is almost 31K for your BASIC programs and the RAM from \$C000 to \$CFFF is left free, since this area is used extensively for machine language programs.

Since Inter Melody uses the system interrupts, it does not work well during input or output. However you can experiment with the idea of playing a song while the computer a program. ■

Before typing this program, read "How to Enter Programs" and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Inter Melody Code

```
10 PRINT"[CLEAR,RIGHT14,RVS]
   INTER MELODY"'BAAD
15 FOR AD=36818 TO 36863:READ BYT
   :POKE AD,BYT:S=S+BYT:NEXT'IFTO
20 FOR AD=679 TO 762:READ BYT
   :POKE AD,BYT:S=S+BYT:NEXT'IBSJ
25 IF S<>15719 THEN PRINT"[DOWN2,RVS]
   ERROR[RVOFF] IN DATA STATMENTS"
   :END'GHXP
100 DATA 120,160,0,132,155,132,158,
    169,144,133,156,169,152,133,159,
    173,20'BOQH
105 DATA 3,73,150,141,20,3,173,21,3,
    73,232,141,21,3,48,7,169'BBPK
110 DATA 15,141,24,212,16,3,140,24,
    212,88,96,0'BMAD
115 DATA 206,249,2,240,3,76,49,234,
    169,16,160,32,141,11,212,140,
    18'BHRM
120 DATA 212,160,0,162,1,177,155,157,
    7,212,177,158,157,14,212,230,
    155'BKQJ
125 DATA 230,158,208,4,230,156,230,
    159,202,16,233,177,155,201,255,
    208,12'BNLO
130 DATA 132,155,132,158,160,144,132,
    156,160,152,132,159,174,250,2,142,
    249'BPJL
135 DATA 2,169,17,160,33,141,11,212,
    140,18,212,76,49,234,1,0'BBPN
```

END

Inter Melody

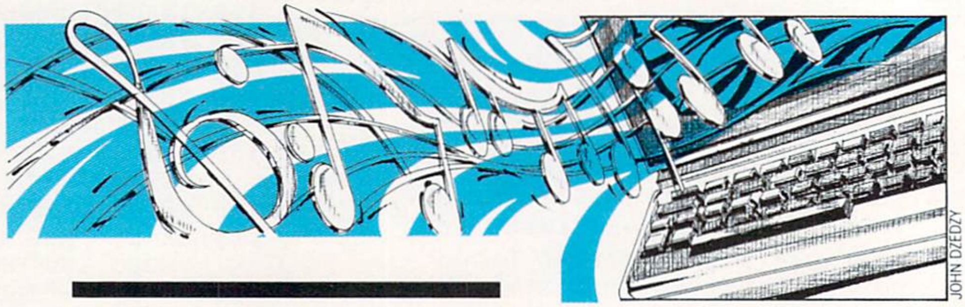
```
100 IF PEEK(679)<>206 THEN PRINT"[RVS]
    STOP:[RVOFF] LOAD AND RUN IM CODE
    FIRST":END'HJOK
110 POKE 55,208:POKE 56,143:CLR'DOGA
120 DIM NT$(12),FR(12):S=54272
    :BASE=36864:T=1:FOR L=S TO S+24
    :POKE L,0:NEXT'KUIN
130 INPUT"[CLEAR,DOWN2]
    ENTER THE TWO VOICE NUMBERS";A,B
    :A=(A-1)*7:B=(B-1)*7'HSOO
140 AD(1)=0:SR(1)=240:AD(2)=9:SR(2)=0
    :PH(1)=0:PL(1)=0:PH(2)=0
    :PL(2)=0'IHDQ
150 WA=16:WB=32:TEMPO=15
    :REM WAVEFORM (16,32,64,
    128) & BEAT (60TH OF SEC)'EGBQ
160 INPUT"[DOWN2]HAS THE DATA BEEN
    PROCESSED (Y,N)";A$:PRINT"[CLEAR]"
    :IF A$="Y"THEN 350'FKFP
170 FOR I=1 TO 12:READ NT$(I),FR(I)
    :NEXT'FSNI
180 C=0:OC=4:REM START AT 4TH
    OCTAVE'DXJL
190 READ C$:IF C$="END"THEN 310'EIYI
200 IF C$="RES"THEN HF%=0:LF%=0
    :GOTO 280'GPCD
210 IF C$="+"THEN OC=OC+1
    :GOTO 190'GLHD
220 IF C$="-"THEN OC=OC-1
    :GOTO 190'GLKE
230 IF VAL(C$)>0 THEN 290'EIRD
240 FOR I=1 TO 12'DEEC
250 IF C$=NT$(I)THEN F=FR(I):I=12'FSII
260 NEXT:F=F/(2^(7-OC))'FLGH
270 HF%=F/256:LF%=F-256*HF%'FSHK
280 POKE BASE+C,HF%:C=C+1
    :POKE BASE+C,LF%:C=C+1
    :PRINT"[SHFT W]";:GOTO 190'KIXT
290 FOR I=1 TO VAL(C$)-1'FHTJ
300 POKE BASE+C,HF%:C=C+1
    :POKE BASE+C,LF%:C=C+1
    :PRINT"[SHFT W]";:NEXT
    :GOTO 190'LJKM
310 H(T)=C:T=T+1:POKE BASE+C,255
    :PRINT'GUVG
320 IF(BASE+C)>=(BASE+2048)THEN
    PRINT"[RVS]ERROR:[RVOFF]
     OUT OF MEMORY":END'ISAO
330 IF T=2 THEN BASE=BASE+2048
    :GOTO 180'GSMI
340 IF H(1)<>H(2)THEN PRINT"[RVS]ERROR
    :[RVOFF] NUMBER OF NOTES DO NOT
    MATCH":END'GJFQ
350 POKE 704,A:POKE 709,B
    :POKE 762,TEMPO'DVII
360 POKE 692,A+4:POKE 753,A+4
    :POKE 688,WA:POKE 749,WA+1
    :REM SET WAVEFORM A'IQIR
370 POKE 695,B+4:POKE 756,B+4
    :POKE 690,WB:POKE 751,WB+1
    :REM SET WAVEFORM B'IQFS
380 POKE S+A+5,AD(1):POKE S+A+6,SR(1)
    :POKE S+B+5,AD(2):POKE S+B+6,
    SR(2)'MOSV
390 POKE S+A+2,PL(1):POKE S+A+3,PH(1)
    :POKE S+B+2,PL(2):POKE S+B+3,
    PH(2)'MOBW
400 PRINT"[DOWN2]SYS(36818) TO TURN
    ON/OFF"'BAEF
996 DATA C,34334,C#,36376, D,
    38539'BYWX
997 DATA D#,40830, E,43258, F,
    45830'BYOY
998 DATA F#,48556, G,51443,G#,
    54502'BAIA
999 DATA A,57743,A#,61176, B,
    64814'BYQB
```

END

BY DONALD J. EDDINGTON

Magic Music Box

for the Commodore 64



Here is a true music processor.

This program can literally process a melody line! You will see it eat the melody, digest it into a variation on the melody, then grind-out the melody. Watch the notes get eaten one by one. Hear and see Magic Music Box noodle-around with the melody notes. Watch and hear the inner workings of the processor box. See the melody, all processed and ready to play, shoot out of the Magic Music Box (last note first) and play for you. Here is a true music processor program.

Type in a melody, using the number keys (or letters for rests in the melody line), and hit RETURN. Or just hit the F1 key and get a melody all ready to play. Now the Magic Music Box is ready to perform its magic for you.

The melody is eaten a note at a time. Each note goes down the melody chute, through the flashing Ex-o-rator, and into the processor box. Once in the processor box, the note is noodled with three times and tested for placement in the finished melody. Each note in the original melody will go through this processing procedure.

When all of the notes are in the box and all of the noodling with and placement of notes is completed, then the music processing is complete. The Magic Music Box now memorizes the melody just processed and prints the variation type on the front of the processor box.

Now that the variation type is listed and all of the processing is done, the melody drop chute opens and the finished melody comes shooting out—last note first. Finally the finished melody is played, using the variation type still listed on the front of the processor box.

What is really happening behind the scenes is a fancy form of string manipulation and printing of altered strings. Also, a lot of color and character poking is occurring, with the SID chip poking as well.

Your melody (found in M$) is a string of numbers (about 75 of them) from zero to nine. Once you press RETURN, this string is printed and a space is added to the front of the string. Then the string is printed again, but only the original LEN$ value is used. This makes the last character of the string not print. The (LEN(A$)) coupled with the LEFT$ will do this—Line 227 is an example. The string is really growing in length (it is double by the end of this process), but the effect of the operation is to make the string shrink! This gives the effect of eating the melody.

The melody chute and the processor box are drawn with standard keyboard graphics. The Ex-o-rator is a graphic with pokes to screen color memory to make it flash.

The processing of the melody is done with rapid pokes to the SID chip. The numbers in the processor box keep changing places and colors. That is, the processing operation is simply more screen and color memory pokes.

The melody drop chute is done with graphic pokes to the screen. The melody that comes shooting out of the processor box is achieved by printing the RIGHT$ of the melody over and over, one more character each time, until the whole melody is printed (the value stored in LEN M$).

To play the melody after it has been processed is just more string manipulation. It starts with the first number in the string and converts it to ASCII code then to screen character code (a difference or 48, of Hex $30). Now, a note array is referenced at this number and the SID chip is poked with the array variables just accessed to produce the correct note. Then, the next number is extracted from the melody string (using the MID$ function), and that note is found and played. This continues until the entire melody string has been played (or LEN M$). After the song has ended, the menu screen returns so that you can hear the same melody in a different variation, try a new melody for input, or end the program.

The variations are just as stated and all are done with more string manipulation. Forward reads the melody string just as you input it—front to back or first to last. Backward reads the string from last note to first note—backward.

To invert a number string, the value of each number (character) in the string is found and subtracted from ten. This will make the low notes be the high notes now, and vice-versa. This turns the melody upside-down—or inverts it.

The inverted and backward operation first inverts the string (described above), then plays it last note to first, or backward. Also note that any invert operations will convert any letters of graphics in the melody string to colons and interpret them as rests in the music.

Your input of the Melody String must be limited to two lines, because that is the limit of the INPUT M$ routine used in the program. You will find this long enough for most melodies. Also, please stick with numbers only in your input string. Letters and/or graphics will be accepted, but only produce rests or silences in the melody when played. They are also converted to colons when any invert operation is selected.

The music system used is the same system used in numbering chord-organ books marked C-Organ, so any piece in these books which is not too long can be input into this program. Also, notes with a plus after them or any note numbers over nine can not be played, so choose carefully. The only thing to watch closely is the note values in these books, needed to make the rhythm of the melody come

out right. Below is a conversion chart for all of the note values and types you will find in the melodies.

Eighth note	:	uses 1 number and looks like this: ♪ or 𝅘𝅥𝅮
Quarter note	:	uses 2 numbers and looks like this: ♩ or ♩
Dotted-quarter	:	uses 3 numbers and looks like this: ♩. or ♩.
Half note	:	uses 4 numbers and looks like this: 𝅗𝅥 or 𝅗𝅥
Dotted-half	:	uses 6 numbers and looks like this: 𝅗𝅥. or 𝅗𝅥.
Whole note	:	uses 8 numbers and looks like this: 𝅝 only

Try these strings for melody string input:

5 5 5 6 5 5 4 4 3 3 4 4 5 5 5 0 2 2 3 3 4 4
4 0 3 3 4 4 5 5 5 0 5 5 5 6 5 5 4 4 3 3 4 4
5 5 5 0 2 2 0 0 5 5 0 0 3 3 1 1 1 1 1 0 2 3
4 3 4 3 2 3 4 3 4 3 2 3 4 3 4 5 6 6 9 6 6 0
5 6 5 4 5 6 9 8 7 6 6 0 5 6 5 4 5 3 2 0 9 2
2 0 3 3 3 0 3 3 2 2 1 1 1 0 1 1 1 0 2 2 2 0
2 2 4 4 3 3 2 2 1 1 1 0 5 5 5 0 5 5 4 4 3 3
3 0 3 3 1 1 2 2 4 4 3 3 2 2 1 1 1 1 1 1 0 0

Remember that the note values in the melody strings are made to play the correct rhythm by using the note value conversion chart. Some conversion examples are half-note G is 5555; whole note C is 11111111 high-D dotted-quarter is 999; eighth note F is 4; and eighth A is 6.

Watch Magic Music Box do its tricks for you. You will get a big surprise when you end the program with selection six! Try processing some melodies today. ■

Before typing this program, read "How to Enter Programs" and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Magic Music Box

```
10 PRINT"[CLEAR,BLUE,DOWN,SPACE5,RVS,
   SPACE2]THIS IS THE COMMODORE 64
   [SPACE2]"'BAFF
20 PRINT"[RVS,RED]"," MAGIC MUSIC BOX
   "'BBYD
30 FOR N=1 TO 9:READ H,L:NH(N)=H
   :NL(N)=L:NEXT'HWPI
40 DATA 16,195,18,209,21,31,22,96,25,
   30,28,49,31,165,33,135,37,162'BIGL
50 POKE 54296,10:POKE 53281,1
   :POKE 54277,47:POKE 54278,12
   :BG=53280:POKE BG,14'GXFO
55 PRINT"[DOWN2,GRAY1,SPACE2]
   THIS PROGRAM TAKES YOUR MELODY,
   "'BAIQ
60 PRINT"AND FEEDS IT INTO THE MAGIC
   MUSIC BOX"'BAIM
65 PRINT" WHICH PROCESSES IT,
    THEN PLAYS IT :"'BAYR
70 PRINT"[SPACE3,GREEN]FORWARDS, [RED]
   BACKWARDS, [BLUE]INVERTED,"'BAPM
75 PRINT"[SPACE5,PURPLE]
   OR BACKWARDS AND[SPACE2]
   INVERTED."'BACR
80 PRINT"[DOWN2,RED]TYPE A KEY TO GO
   ON, OR TYPE [RVS]F1[RVOFF]
    FOR DEMO"'BAXP
85 M$="4Ø45Ø56864Ø14Ø4545666401404505
   68644Ø9ØØ5Ø866644"'BCPT
90 GET A$:IF A$=""GOTO 9Ø'EHLI
95 IF A$="[F1]"THEN DEMO=1
   :GOTO 2ØØ'FLSQ
100 PRINT"[CLEAR,DOWN2,BLACK,SPACE2]
    NOW, TYPE IN A SERIES OF
    NOTES"'BABF
105 PRINT"[SPACE4]USING THIS[SPACE2]
    SYSTEM:[DOWN]"'BASG
110 PRINT"1=MID-C","2=D","3=E","4=F"
    :PRINT"5=G","6=A","7=B",
    "8=HI-C"'CHDG
115 PRINT"9=HI-D","Ø=REST"'BBSE
120 PRINT"[DOWN2]TYPE YOUR MELODY,
     THEN THE [RVS]RETURN[RVOFF]
     KEY"'BAII
125 PRINT"[RED,RVS] LESS THAN TWO
    LINES, PLEASE! [BLACK,RVOFF]"
    :INPUT M$'CDHM
130 PRINT"[CLEAR,DOWN2,RED] NOW,
     SELECT THE WAY YOU WANT IT
    PLAYED FROM THE FOLLOWING:"'BANP
140 PRINT"[DOWN2,GREEN] 1=FORWARDS"
    :PRINT"[RED] 2=BACKWARDS"
    :PRINT"[BLUE] 3=INVERTED"'DCYK
150 PRINT"[PURPLE] 4=BOTH 2 AND 3"
    :PRINT"[BLACK] 5=INPUT NEW
    MELODY"'CBHL
155 PRINT"[BROWN] 6=END PROGRAM[BLACK]
    "'BAJK
160 INPUT W$:W=VAL(W$)'DILE
165 IF W>6 OR W<Ø THEN PRINT"I'LL
    ASSUME YOU WANTED FORWARDS"
    :W=1'HHAV
170 FOR D=1 TO 1ØØØ:NEXT'EHAF
180 ON W GOTO 2ØØ,3ØØ,4ØØ,5ØØ,1ØØ,
    6ØØ'CYSI
200 REM PLAY THE MELODY
    **-FORWARDS-**'BCRD
210 PRINT"[CLEAR,DOWN2]","[SPACE2,RVS,
    GREEN,SPACE2]FORWARDS[SPACE3,
    DOWN2]":POKE BG,5:GOSUB 7ØØ'DKDF
220 PRINT,"[DOWN,BROWN]
     IT GOES IN HERE!![GRAY1]"
    :PRINT M$:FOR D=1 TO 5ØØ
    :NEXT D'GMHJ
226 A$=M$:F$=M$:PRINT"[HOME,DOWN7]";
    A$;'DOJJ
227 P=LEN(A$)-1:FOR T=1 TO LEN(A$)
    :A$=" "+LEFT$(A$,P)+" [SHFT N]
    "'LWUS
228 PRINT"[HOME,DOWN7]";A$;
    :FOR D=1 TO 34:NEXT D
    :FOR K=1 TO 3:GOSUB 8ØØ
```

```
    :NEXT K,T'KXTR
229 PRINT,"[HOME,DOWN10,RIGHT15,GREEN]
    FORWARDS[DOWN]":POKE 1344+I,32
    :GOSUB 900'EOJR
230 FOR T=0 TO LEN(M$):T$=RIGHT$(M$,T)
    :PRINT"[HOME,DOWN13]"SPC(8)"
    [SHFT M] "T$'IUBK
231 FOR D=1 TO 50:NEXT:NEXT
    :FOR D=1 TO 500:NEXT'JNAI
235 FOR Q=1 TO LEN(M$)'EGVH
240 G=(ASC(MID$(M$,Q,1))-48)'EPVF
250 IF G<0 OR G>9 THEN G=0'GGBG
260 POKE 54273,NH(G):POKE 54272,NL(G)
    :POKE 54276,33'DHWK
270 FOR D=1 TO 77:NEXT D:POKE 54276,32
    :NEXT Q'GRXK
280 FOR X=1 TO 1234:NEXT:POKE 54273,0
    :POKE 54272,0:IF DEMO THEN
    RUN'JDYP
290 GOTO 130'BDDG
300 REM PLAY THE MELODY
    **-BACKWARDS-**'BDEF
310 PRINT"[CLEAR,DOWN2]","[SPACE2,RVS,
    RED,SPACE2]BACKWARDS![SPACE2,
    DOWN2]":POKE BG,2:GOSUB 700'DKRG
320 PRINT,"[DOWN,BROWN]
     IT GOES IN HERE!![GRAY1]"
    :PRINT M$:FOR D=1 TO 500
    :NEXT D'GMHK
326 A$=M$:FOR T=LEN(M$)TO 1 STEP-1
    :B$=B$+MID$(M$,T,1):NEXT
    :PRINT"[HOME,DOWN7]"A$;'MFVT
327 P=LEN(A$)-1:FOR T=1 TO LEN(A$)
    :A$=" "+LEFT$(A$,P)+" [SHFT N]
    "'LWUT
328 PRINT"[HOME,DOWN7]";A$;
    :FOR D=1 TO 34:NEXT D
    :FOR K=1 TO 3:GOSUB 800
    :NEXT K,T'KXTS
329 PRINT,"[HOME,DOWN10,RIGHT14,RED]
    BACKWARDS[DOWN]":POKE 1344+I,32
    :GOSUB 900'EOVS
330 FOR T=0 TO LEN(M$):T$=RIGHT$(B$,T)
    :PRINT"[HOME,DOWN13]"SPC(8)"
    [SHFT M] "T$'IUPL
331 FOR D=1 TO 50:NEXT:NEXT
    :FOR D=1 TO 500:NEXT'JNAJ
335 FOR Q=LEN(M$)TO 1 STEP-1'GHXK
340 G=(ASC(MID$(M$,Q,1))-48)'EPVG
350 IF G<0 OR G>9 THEN G=0'GGBH
360 POKE 54273,NH(G):POKE 54272,NL(G)
    :POKE 54276,33'DHWL
370 FOR D=1 TO 77:NEXT D:POKE 54276,32
    :NEXT Q'GRXL
380 FOR X=1 TO 1234:NEXT:POKE 54273,0
    :POKE 54272,0:GOTO 130'HCJO
400 REM PLAY THE MELODY
    **-INVERTED-**'BCKF
410 PRINT"[CLEAR,DOWN2]","[SPACE2,RVS,
    BLUE] INVERTED!! [DOWN2]"
    :POKE BG,6:GOSUB 700'DKTH
420 PRINT,"[DOWN,BROWN]
     IT GOES IN HERE!![GRAY1]"
    :PRINT M$:FOR D=1 TO 500
    :NEXT D'GMHL
426 A$=M$:FOR T=1 TO LEN(M$)
    :S=10-(VAL(MID$(M$,T,1)))
    :I$=I$+CHR$(S+48):NEXT'ONGX
427 PRINT"[HOME,DOWN7]"A$;:P=LEN(A$)-1
    :FOR T=1 TO LEN(A$)
    :A$=" "+LEFT$(A$,P)+" [SHFT N]
    "'MBRW
428 PRINT"[HOME,DOWN7]";A$;
    :FOR D=1 TO 34:NEXT D
    :FOR K=1 TO 3:GOSUB 800
    :NEXT K,T'KXTT
429 PRINT,"[HOME,DOWN10,RIGHT15,BLUE]
    INVERTED[DOWN]":POKE 1344+I,32
    :GOSUB 900'EODT
430 FOR T=0 TO LEN(M$):T$=RIGHT$(I$,T)
    :PRINT"[HOME,DOWN13]"SPC(8)"
    [SHFT M] "T$'IUWM
431 FOR D=1 TO 50:NEXT:NEXT
    :FOR D=1 TO 500:NEXT'JNAK
435 FOR Q=1 TO LEN(M$)'EGVJ
440 G=10-(VAL(MID$(M$,Q,1)))'EPJH
460 POKE 54273,NH(G):POKE 54272,NL(G)
    :POKE 54276,33'DHWM
470 FOR D=1 TO 77:NEXT D:POKE 54276,32
    :NEXT Q'GRXM
480 FOR X=1 TO 1234:NEXT:POKE 54273,0
    :POKE 54272,0:GOTO 130'HCJP
500 REM PLAY THE MELODY **-INVERTED
    AND BACKWARDS-**'BOFK
510 PRINT"[CLEAR,DOWN2]","[LEFT3]
    [RVS,PURPLE] INVERTED & BACKWARDS
    [DOWN2]":POKE BG,4:GOSUB 700'DKJN
520 PRINT,"[DOWN,BROWN]
     IT GOES IN HERE!![GRAY1]"
    :PRINT M$:FOR D=1 TO 500
    :NEXT D'GMHM
526 A$=M$:FOR T=LEN(M$)TO 1 STEP-1
    :S=10-(VAL(MID$(M$,T,1)))'LDQU
527 IB$=IB$+CHR$(S+48):NEXT'FMVN
528 PRINT"[HOME,DOWN7]"A$;:P=LEN(A$)-1
    :FOR T=1 TO LEN(A$)
    :A$=" "+LEFT$(A$,P)+" [SHFT N]
    "'MBRY
529 PRINT"[HOME,DOWN7]";A$;
    :FOR D=1 TO 34:NEXT D
    :FOR K=1 TO 3:GOSUB 800
    :NEXT K,T'KXTV
530 PRINT,"[HOME,DOWN10,RIGHT13,
    PURPLE]INVERT & BACK[DOWN]"
    :POKE 1344+I,32:GOSUB 900'EOGN
531 FOR T=0 TO LEN(M$):T$=RIGHT$(IB$,
    T):PRINT"[HOME,DOWN13]"SPC(8)"
    [SHFT M] "T$'IVNO
532 FOR D=1 TO 50:NEXT:NEXT
    :FOR D=1 TO 500:NEXT'JNAM
535 FOR Q=LEN(M$)TO 1 STEP-1'GHXM
540 G=10-(VAL(MID$(M$,Q,1)))'EPJI
```

Continued on pg. 123

BY BENNETT COOKSON, JR.

Instant LOAD and Directory

for the Commodore 64

Add a full-featured directory and disk command to BASIC.

Instant LOAD and Directory adds a full-featured directory and disk command to BASIC. At the touch of a key, any program in the directory will be loaded and run. The DIR command has options for a dual drive, drive #9, and even extra long directories. Both commands can also be used within a program.

The Directory

DIR[d][,device number]

The above d is optional and can be a drive number or string as in the following examples.

DIR	:rem list disk directory
DIR1	:rem directory from dual drive 1
DIR"$0:P*"	:rem directory of programs that start with P
DIR,9	:rem directory from device number 9
DIR1,9	:rem directory from dual drive #1, device #9

Often a long directory scrolls off the screen before you have a chance to read it. To solve the problem, add an exclamation mark before the DIR command and the computer will wait for you to press a key before listing each line of the directory. The exclamation mark can be added before any DIR command. For example,

!DIR
!DIR"$1:E*",9

The Disk Command

The DISK command is used to send commands to the disk drive and read the error channel. See your disk drive manual for a list of disk commands.

DISK["disk command"][,device number]

DISK"I0:"	:rem initialize drive 0
DISK	:rem read disk error channel
DISK,9	:rem read error from disk drive #9
DISK"V0:",9	:rem send command to device #9

Using Dir and Disk within a Program

So that BASIC will not be slowed, the computer only checks for the DIR and DISK commands in direct mode. However, you can still use these commands in a program line with the help of the SYS command. SYS 49183 must be the statement before each DIR or DISK command. For example,

10 SYS 49183:DIR:SYS 49183:DISK"I0:"

The SYS tells the computer to check the next statement for a DIR or DISK command.

You can now use the function keys to instantly load any program from the directory. Simply list the directory on the screen and move the cursor up the left side of the screen to the program you want to load and press F1. Your program will then be loaded and automatically run.

On the screen, a directory entry will look like this.

15 "FAVORITE PROGRAM" PRG

All we need to make a LOAD command out of the above directory line is to add the keyword LOAD and a ",8" at the end. This is what happens when you press F1. In addition, a return and the keyword RUN are placed in the keyword buffer so the program will start automatically. The other function keys are used for other load options and a directory.

F1 = LOAD	,8:	RUN
F3 = LOAD	,8,1:	
F5 = LOAD	,8:	
F7 = !DIR		

Notice that F1 is the only one that automatically runs a program, F3 is for machine-language programs, and F7 is a quick way to get a directory. Also note that Instant LOAD and Directory will only work in direct mode. (Sometimes when you break out of a program, you will need to use a command like PRINT to set the flags for direct mode.)

Before you can use Instant LOAD and Directory you will need to load it. However, once in memory all other load will be done instantly!

LOAD"DIR",8
RUN

(Use SYS 49152 to restart function keys after RUN/STOP-RESTORE.)

Before typing this program, read "How to Enter Programs" and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Instant Load and Directory

```
10 POKE 53280,0:POKE 53281,0'CPLB
20 PRINT CHR$(147):PRINT'DGGB
30 PRINT TAB(7)"INSTANT LOAD AND
   DIRECTORY"'CCTH
40 PRINT:PRINT TAB(9)"BY BENNETT
   COOKSON JR.":PRINT:'EFEJ
50 PRINT "COMMANDS"'BARD
60 PRINT CHR$(30):PRINT"DIR[SPACE6]
   :REM LIST DIRECTORY"'DFHL
70 PRINT"!DIR[SPACE5]:REM SLOWLY LIST
   DIRECTORY"'BAVM
80 PRINT"DISK[SPACE5]:REM READ DISK
   ERROR"'BAYL
90 PRINT"DISK";CHR$(34);"S0
   :PRG";CHR$(34);"[SPACE3]
   :REM SEND DISK COMMAND"'DMMS
100 PRINT:PRINT:PRINT"FUNCTION
    KEYS"'DCNB
110 PRINT"(PLACE CURSUR ON PROGRAM TO
    LOAD)"'BACF
120 PRINT:PRINT"F1 : LOAD & RUN
    PROGRAM"'CBIE
130 PRINT"F3 : LOAD ML PROGRAM (,8,
    1)"'BADE
140 PRINT"F5 : LOAD PROGRAM (NO
    RUN)"'BAEG
150 PRINT"F7 : !DIR"'BAYC
160 PRINT:PRINT:PRINT"READING
    DATA...";'DDTH
170 FOR I=49152 TO 49768:READ A
    :POKE I,A:C=C+A:NEXT'IWQL
180 IF C<>74434 THEN PRINT"ERROR IN
    DATA, CHECK FOR TYPOS":END'GHDQ
```

```
185 FOR X=1 TO 7:READ A:POKE 630+X,A
    :NEXT'HNUP
189 POKE 198,7'BFFN
190 SYS 49152:END'CGRG
200 DATA 120,169,23,160,192,141,8,
    3'BBFB
210 DATA 140,9,3,169,191,160,193,
    141'BCNC
220 DATA 143,2,140,144,2,88,96,
    166'BARD
230 DATA 58,232,240,3,76,228,167,
    160'BCSE
240 DATA 1,140,104,194,177,122,201,
    33'BDUF
250 DATA 208,8,169,240,141,104,194,
    200'BEAH
260 DATA 177,122,201,68,208,230,200,
    177'BFAI
270 DATA 122,201,73,208,223,200,177,
    122'BFNJ
280 DATA 201,82,208,6,32,91,192,
    76'BAQJ
290 DATA 174,167,201,83,208,206,200
    177'BFGL
300 DATA 122,201,75,208,199,32,79,
    193'BDYC
310 DATA 76,174,167,32,115,0,136,
    208'BCOD
320 DATA 250,44,104,194,80,7,169,
    40'BBOE
330 DATA 160,194,32,25,194,32,204,
    255'BDLF
340 DATA 169,120,32,195,255,32,115,
    0'BCGG
350 DATA 240,33,32,158,173,36,13,
    48'BBMH
360 DATA 16,32,161,183,224,1,208,
    19'BBGI
370 DATA 169,2,162,102,160,194,76,
    161'BDNJ
380 DATA 192,32,163,182,166,34,164,
    35'BDSK
390 DATA 76,161,192,169,2,162,100,
    160'BDJL
400 DATA 194,32,189,255,32,121,0,
    240'BCHD
410 DATA 4,32,155,183,44,162,8,
    169'BAUE
420 DATA 120,160,0,32,186,255,32,
    192'BCCF
430 DATA 255,144,4,170,76,55,164,
    32'BBRG
440 DATA 204,255,162,120,32,30,225,
    32'BDSH
450 DATA 140,193,32,36,225,32,36,
    225'BCHI
460 DATA 32,36,225,240,34,32,36,
    225'BBEJ
470 DATA 170,32,36,225,32,205,189,
    169'BDSK
480 DATA 32,32,210,255,32,36,225,
    32'BBXL
490 DATA 210,255,208,248,169,13,32,
    210'BEAN
500 DATA 255,32,250,192,76,205,192,
    76'BDWE
510 DATA 182,193,32,204,255,44,104,
    194'BEHG
520 DATA 80,16,169,0,133,198,165,
    145'BCVG
530 DATA 16,17,165,198,240,248,169,
    0'BCYH
540 DATA 133,198,165,145,16,5,162,
    120'BDMI
550 DATA 76,30,225,32,182,193,56,
    76'BBXJ
560 DATA 75,168,32,121,0,201,44,
    208'BBCK
570 DATA 4,32,155,183,44,162,8,
    134'BAML
580 DATA 87,32,204,255,169,120,32,
    195'BDSM
590 DATA 255,169,0,32,189,255,169,
    120'BDXN
600 DATA 166,87,160,15,32,186,255,
    32'BCVF
610 DATA 192,255,144,63,76,187,192,
    32'BDEG
620 DATA 115,0,136,208,250,32,115,
    0'BBQH
630 DATA 240,68,201,44,240,70,32,
    158'BCGI
640 DATA 173,32,163,182,72,165,34,
    72'BCRJ
650 DATA 165,35,72,32,34,193,162,
    120'BCJK
660 DATA 32,24,225,104,133,35,104,
    133'BDSL
670 DATA 34,104,170,160,0,177,34,
    32'BBYM
680 DATA 210,255,200,202,208,247,76,
    204'BFTO
690 DATA 255,165,215,96,32,36,225,
    133'BDSO
700 DATA 215,32,183,255,240,243,104,
    104'BFSH
710 DATA 165,186,201,8,208,24,32,
    45'BBNH
720 DATA 193,76,167,193,32,34,193,
    162'BDDI
730 DATA 120,32,30,225,32,36,225,
    32'BBSJ
740 DATA 210,255,201,13,208,246,32,
    204'BEOL
750 DATA 255,169,120,32,195,255,96,
    166'BEYM
760 DATA 58,232,208,10,165,203,197,
    197'BEPN
770 DATA 240,4,201,7,144,3,76,72'BXLM
780 DATA 235,201,3,144,249,133,197,
    201'BEAP
790 DATA 3,208,10,169,88,160,194,
```

INSTANT LOAD / DIRECTORY

```
     32'BBRP
800 DATA 25,194,76,1,194,166,214,
     160'BCTH
810 DATA 0,24,32,240,255,169,56,
     160'BBII
820 DATA 194,32,25,194,165,203,201,
     5'BCIJ
830 DATA 208,7,169,83,160,194,32,
     25'BBWK
840 DATA 194,162,1,165,203,201,4,
     208'BCBL
850 DATA 2,162,6,134,198,189,93,
     194'BBHM
860 DATA 157,118,2,202,208,247,76,
     66'BCUN
870 DATA 235,133,189,132,190,160,255,
     200'BGBP
880 DATA 177,189,32,210,255,208,248,
     96'BEAQ
890 DATA 80,82,69,83,83,32,65,78'BXNQ
900 DATA 89,32,75,69,89,13,13,0'BWIH
910 DATA 76,79,65,68,29,29,29,29'BXZJ
920 DATA 29,29,29,29,29,29,29,29'BXPK
930 DATA 29,29,29,29,29,29,29,44'BXML
940 DATA 56,58,0,157,44,49,58,0'BWFL
950 DATA 147,33,68,73,82,0,13,82'BXXM
960 DATA 85,78,58,13,36,48,36,49,
     1'BAHO
970 DATA 144,145,145,145,63,30,13'BYIP
```

END

QUANTUMLINK

Continued from pg. 22

interest.

If you're interested in sports, you can share comments with others in the Sports Center. You'll find Pro Picks during each season or drop by the Grandstand to discuss various popular sports any time of the year.

The Fun & Games area provides instructions for the on-line Q-link games and tournaments. You'll also find Trivia Quizzes, or you can share jokes in The Funny Bone.

The Treasure section is where you can order the video tape or book for clues to finding the buried treasure. You'll also find information about the treasure and rules for the treasure hunt. There's even a message board where users are sharing clues with each other along with helpful hints.

The Rocklink section of Just For Fun is probably one of the more exciting areas. There are theme conferences every night, plenty of music reviews, a top-sellers chart, industry news and hot headlines, plus the daily news beat. They're even starting to hold live conferences with various rock stars on-line. There were three in one month back in November, with more being scheduled all the time. You can check the Best Bets—Don't Miss section of Customer Service for announcements of upcoming Rocklink conferences.

Just after writing this column I'll be off to Toronto for the World of Commodore show, gathering new product information for my section of CIN. I was fortunate enough to make this show, but my full-time work schedule doesn't allow me to make it to every show where information on Commodore products is likely to appear. If you happen to attend a show or club gathering where new products are shown or discussed, why not drop by the New Product Information area and let everyone else know what's coming?

Looks like that's it for this month. I'm trying to get some additional information on *Habitat* as well as whatever hints of new features or services are on the horizon. In the mean time, let me know if you have any questions or suggestions about this column. C

Bob Baker is in charge of the New Products Information area on the QuantumLink network. He can be reached on Q-Link via E-mail addressed to RBAKER.

SILICON VALLEY

Continued from pg. 18

Amiga's resolution modes. Also included is a font mover utility that will let you install the new fonts on to any disk.

Datasoft has finished the **Video Title Shop** for the 64. This program lets you create professional looking titles for your home video tapes. You can use it to add credits, titles and introductions to your home productions, giving them a polished personal touch.

Westcom Industries has finally released **HardHat** for the Amiga. This hard disk backup utility is an essential part of the data protection routine that any business professional should be following. If you've got 10, 20 or more megabytes of data residing on a hard disk drive with no backup, you're flirting with danger. **HardHat** uses data compression techniques to store data on floppies in less space, and allows for incremental or single file backup and recovery. A full catalog of all files and directories is created, and can be searched or listed. The restore process recovers all backed-up files, and includes filenotes as well as protection flags.

Ultimate Media, Inc. has come out with two programs for the 64 that will be of interest to musicians. The first is **Samplescope**, a MIDI sampler program that works with the Akai S612 MIDI sampler. It lets you use your 64 to visually edit samples from the Akai, and lets you save samples to your disk drive. It also adds automatic gain control and splice point cross fade capability to your sampler. The second is **Autopilot**, a MIDI patch editor for the DX-7. This program lets you edit patches, load and save them from a disk, and generate random patches. You can transfer 1 patch or all 32 at once. The entire program is joystick-driven.

From **JDK Images** comes **Pro Video CGI** for the Amiga. This is a professional level video character generator that operates in the high-resolution interlaced mode. It offers a full range of wipes and fades, a variety of font sizes and styles, on-screen composing, status and command line operation. I took one look at this program and finally saw the first program that truly offers a unique reason to use the Amiga on a regular basis in a professional video environment.

That wraps it up for this month. Next month I plan to take an in-depth look at several new games, including a hot new flight simulator, and perhaps a hands-on report on a desktop publishing program for the Amiga. C

BY JERRY A. STURDIVANT

Printer Maze
for the Commodore 128



ART BAXTER

Printer Maze draws symmetrical maze patterns on your screen. Then it expands and sends them, full page, to your printer. For a simple program, it draws some very unique designs. In fact, my kids enjoy coloring in the drawings. They also like the random design because of the maze effect.

This program is made for the 1902 80-column monitor and Commodore's MPS-1000 printer. You don't have to have a printer, but it's nice to have a hard copy of the designs. With some changes, the program will work on the PET, 64 with 40-column monitor and 1525 printer, but you will lose some of its attributes and printer speed.

The program first asks if you want your design to go to the printer or to the screen first. You then have a choice of a regular pattern, mixed pattern or random design.

For random design, the complete printout will be random. If you select regular pattern or mixed pattern, the computer will select four numbers for a design. Or you have the option of putting in your own four numbers (zero through three).

After a design appears on the screen, you can send it to your printer or you may select another design. When the design is printing on the screen, it is preceded by the randomly selected number which designed it. The number is memorized so you won't get the same design twice. If you wish to repeat the design in the future, you need only refer to the letter (P for pattern, M for Mixed) and the number. Random design doesn't matter, it will always be different.

The designs in lines 60 to 80 are four square nine-pixel designs side by side, and is all that is used in the program. Each nine-pixel design is open on the center of the top and sides. This allows them to connect for continual flow designs. Lines 120 to 210 are your input selections.

This program uses the same routine for the screen and the printer. Line 240 is for the printer, setting up the numbers necessary to fill the paper. The printer is device #4. In line 250, if we are sending our designs to the screen (device #3), we only use 39 columns on a 40-column display and 79 for an 80-column screen. Line 270 tells the printer we're going to draw graphics and to make smaller line feeds.

The loop at line 330 picks four random numbers. Line 370 makes sure they're not all the same number. The loop at 410 memorizes the numbers, and at line 460 we check to be sure we haven't used them before. If it's not going to be a completely random design, line 510 will print the pattern number to the screen in case we like it and want to use it again.

Our main printing loop starts at line 540. Line 700 allows us to stop in the middle of a printing, either of the screen or the printer. Touch the spacebar once just as it's finishing a line.

After a screen has been designed, you are given the choice of another design or making a hard copy. When that's finished printing, you can make another one, go for another design, or quit. When you wish to quit be sure to hit the N key, which resets your printer to its original spacing.

Here are some input numbers to try: MIX 1213, MIX 3101, PATTERN 3220, PATTERN 3211.

Experiment with the program. For a different type of design, change line 550 to IF MIX THEN R(5)=R(2):R(2)=R(4):R(4)=R(1):R(1)=R(3):R(3)=R(5).

For an additional set of unusual designs, change the number 4 in line 570 to a 5. You won't be able to preview designs before they go to the printer, but you will get some unusual designs. If you like the design, remember the number and use it again.

If you only have a 40-column screen, make the following changes. In line 250, make ACROSS equal 39 rather than 79. This is for the width of your screen. Careful—don't change line 240. Add the following line: 625 IF SCREEN=1 THEN A=A+1.

If you have a 1525 printer, make the following changes. In line 240, change the number 66 to 49. Remove line 270 (to remove, type the number by itself and hit RETURN). Remove line 660. Change line 680 to: 680 PRINT#1, CHR$(15) P$(X,Y);. Add a line: 695 PRINT#1, CHR$(8). Remove line 760. In line 790, remove one cursor up from the end of the line. Line 830 should just be: 830 CLOSE 1.

Continued on next pg.

128 USERS ONLY/PRINTER MAZE

Before typing this program, read "How to Enter Programs" and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Printer Maze

```
20 REM   INITIALISE'BKVB
30 REM" [CMDR U12]"'BAPI
40 DIM P$(78,3),D$(3),R(5),
   CK$(50)'BCAF
50 CHOICE$(0)="PATTERN "
   :CHOICE$(1)="MIX ":R=RND(-TI)
   :N=1'GFUP
60 D$(1)="[SHFT @,CMDR @,SHFT L,
   SHFT @] [SHFT L] [SHFT M2,SHFT N2]
   "'BFJK
70 D$(2)="[SPACE3,CMDR M] [CMDR G,
   SHFT M,SHFT SPACE,SHFT M,SHFT N,
   SHFT SPACE,SHFT N]"'BFFK
80 D$(3)="[SHFT P,CMDR T,SHFT O,
   SHFT P,SHFT SPACE,SHFT O,SHFT M2,
   SHFT SPACE2,SHFT N2]"'BFKO
90 REM    INPUTS'BGNH
100 REM" [CMDR U8]"'BASB
110 MIX=0:RAN=0:SCREEN=0:NU=0:A=0'FYVF
120 PRINT"[CLEAR,DOWN5]
    (S)CREEN OR (H)ARD COPY"'BAHE
130 GET A$:IF A$="" THEN 130'EIDC
140 IF A$="S" THEN SCREEN=1'EJNE
150 PRINT"[DOWN2](P)ATTERN OR (M)IX
    OR (R)ANDOM"'BALI
160 GET A$:IF A$="" THEN 160'EIGF
170 IF A$="M" THEN MIX=1'EGDG
180 IF A$="R" THEN RAN=1:GOTO 240'FKJI
190 PRINT"[DOWN2]INPUT A NUMBER?
    (Y/N)"'BATK
200 GET A$:IF A$="" THEN 200'EIBA
210 IF A$="Y" THEN GOSUB 870
    :REM GET SELECTED NUMBER'FXMG
220 REM   SET OUTPUT'BJHB
230 REM" [CMDR U12]"'BAPI
240 ACROSS=78:VERTICAL=(66-2)*2
    :DEVICE=4:A=0'GJOM
250 IF SCREEN THEN ACROSS=79
    :VERTICAL=24:DEVICE=3'FION
260 CLOSE 1:OPEN 1,DEVICE'CKHF
270 IF DEVICE=4 THEN CLOSE 6
    :OPEN 6,4,6:PRINT#6,CHR$(18)'HVKM
280 IF NU=1 THEN 540'DGFH
290 IF A$="H" THEN 540'DFRI
300 A$=""'BCXX
310 REM   PICK 4 RND NUMBERS'BPIC
320 REM" [CMDR U20]"'BAJO
330 FOR I=1 TO 4'DDHC
340 R(I)=INT(RND(0)*4)'EKQF
350 HO$(I)=STR$(R(I))'CMIG
360 NEXT'BAED
370 IF R(1)=R(2) AND R(2)=R(3) AND
    R(3)=R(4) THEN 330'HCKO
380 N=N+1:IF N=50 THEN 830
    :REM QUIT'GPKM
390 REM   MEMORIZE THE NUMBERS'BSSL
400 REM" [CMDR U22]"'BAXO
410 FOR I=1 TO 4'DDHB
420 CK$(N)=CK$(N)+RIGHT$(HO$(I),
    1)'DWBG
430 NEXT'BAEB
440 REM   SEE IF USED BEFORE'BPWG
450 REM" [CMDR U20]"'BAJS
460 COMPARE=N'BIHH
470 FOR I=1 TO N-1'EEWI
480 IF CK$(I)=CK$(COMPARE) THEN
    330'DVHM
490 NEXT'BAEH
500 PRINT"[CLEAR]";'BBDA
510 IF RAN=0 THEN PRINT"SELECTION
    NUMBER: " CHOICE$(MIX) CK$(N)'EWEM
520 REM   SET UP THREE LINES'BPTG
530 REM" [CMDR U20]"'BAJR
540 FOR ROW=1 TO VERTICAL/3'ENHJ
550 IF MIX THEN R(5)=R(4):R(4)=R(3)
    :R(3)=R(2):R(2)=R(1)
    :R(1)=R(5)'HWVS
560 FOR X=1 TO ACROSS/3'EJLK
570 A=A+1:IF A>4 THEN A=1'GIVL
580 R=R(A)*3+1'DHLK
590 IF RAN THEN R=INT(RND(0)*4)
    *3+1'IMRP
600 FOR Y=1 TO 3'DDWC
610 P$(X,Y)=MID$(D$(Y),R,3)'CSSG
620 NEXT Y,X'BDTD
630 REM   PRINT THOSE THREE LINES'BUUJ
640 REM" [CMDR U25]"'BAQW
650 FOR Y=1 TO 3'DDWH
660 PRINT#1'BBAG
670 FOR X=1 TO ACROSS/3'EJLM
680 PRINT#1,P$(X,Y);'BKJK
690 NEXT X'BBRJ
700 GET A$:IF A$=" " THEN 730'EIQF
710 NEXT Y'BBSC
720 NEXT ROW'BDVE
730 IF SCREEN=0 THEN FOR I=1 TO 6
    :PRINT#1:NEXT'INAM
740 REM   FULL PAGE-NOW WHAT?'BROK
750 REM" [CMDR U21]"'BATV
760 PRINT'BACH
770 PRINT"[SPACE2]MORE? (Y/N) HARD
    COPY? (H) * ";'BBXP
780 IF RAN=0 THEN PRINT
    LEFT$(CHOICE$(MIX),1) CK$(N);'FCHS
790 PRINT " *[UP2]"'BAIM
800 GET A$:IF A$="" THEN 800'EIHG
810 IF A$="H" THEN SCREEN=0
    :GOTO 240'FNPJ
820 IF A$="Y" THEN 110'DFCH
830 CLOSE 1:IF DEVICE=4 THEN PRINT#6,
    CHR$(36):CLOSE 6'HRSN
840 END'BACG
850 REM   INPUT SELECTED NUMBER'BTXN
860 REM" [CMDR U23]"'BACA
870 N=N+1:NU=1'DHHM
880 INPUT "[DOWN]ENTER THE 4 NUMBERS";
    CK$(N)'BHOR
890 FOR I= 1 TO 4'DDHN
900 R(I)=VAL(MID$(CK$(N),I,1))'DSQI
910 NEXT'BAEE
920 RETURN'BAQF
```

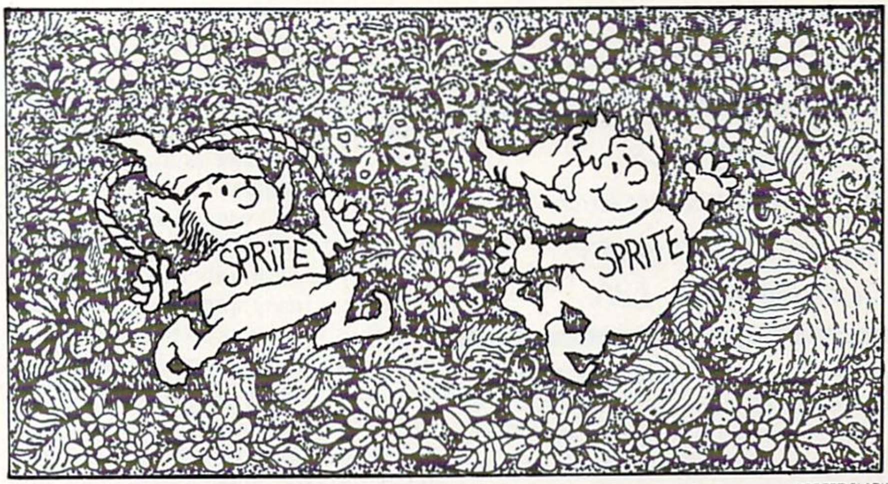
END

BY MARK JORDAN

The 128 Mode

Smoke-Busters

Explore the Commodore 128 in its powerful native mode. Some of the articles in this column may be technical, some not so technical—but we guarantee that they will spark your creativity.



ROBERT CLARK

I'll start this month by raving about the 128, the top eight-bit computer in the world. Though some of my contemporaries are proclaiming the twilight of the eight-bit era, I think they are premature. This machine has a BASIC that is fully loaded, a crisp 80-column display that is also capable of 640 x 200 pixel high-resolution graphics, a sophisticated memory manager that allows up to 640K of memory, and a screen editor that is possibly the best of any personal computer.

Big statements, all. But over the next months if you journey with me, you will be convinced. Let me start backing up my words right now by showing how the 128 takes an existing technology to its ultimate. The technology I'm talking about is sprites.

If you've ever glanced at your system guide, you've undoubtedly heard of sprites. You may know that they are 24 television dots (called pixels) wide by 21 long. You may even know that their name comes from the word spirits. (Shakespeare uses the term sprites several times). But you may be wondering what's so wonderful about them.

Here's what. They are controlled by a separate piece of silicon in your computer called the VIC (for Video Interface Chip). It's wonderful because it does all kinds of messy calculations for you the programmer. It keeps track of which sprite should appear on top of which, revealing the overlaid sprite just like real-life 3D. It keeps track of when two sprites collide. It remembers and restores whatever was on the screen before a sprite passed through. The Commodore 64 has sprites, the eight-bit Ataris have a form of sprites, and the now-defunct TI-99 4/A had them too.

The thing that makes sprites stand out on the 128 is the way they are implemented through the operating system and BASIC. Anyone who is willing to read the system guide carefully can have sprites up and running in no time flat. The built-in SPRDEF command jumps you into a very useful sprite editor. While not as sophisticated as some that have been written for the 64, most of what you need is there. The important thing is, SPRDEF is always there, waiting to be used. It can even be part of a running program. (Next month we'll do just that.)

After you've designed sprites, you can move sprites simply by turning them on with SPRITE and using MOVSPR. I won't waste space explaining what the system guide covers so adequately.

If you like your creations, you can save them painlessly with the wonderful (very, very wonderful) BSAVE command. This little gem lets you save your sprites without getting your hands dirty with data.

Then, too, there are times when you'll need to dirty your program up with data. The 128's built-in machine-language monitor allows you to do a hexadecimal memory dump (which BASIC 7.0 has no problem interpreting) which, with the aid of some nifty re-programmed function keys, can be transformed into data statements in about two minutes.

Other commands such as BUMP, COLLISION and RSSPOS simplify sprite work greatly. But the best, the very best of all, has not been mentioned: SPRSAV (and his trusty sidekicks, SSHAPE and GSHAPE). When the engineers sneaked this one in, they put a seal on the now famous 24 x 21 grid of pixels that comprise a sprite's shape. Here's what I mean.

In Smoke-Busters I decided I wanted to have six sprites in the shape of peoples' faces, so they could be looking out the windows of an apartment building. I wanted these faces to be hidden in the dark, only to be revealed when the spotlight from the smoke-buster's gun shined on them.

That much was no problem—I could simply use sprite seven or eight for the spotlight and sprites one through six for the faces because, according to VIC, the lower the number of sprite, the higher the priority it has in being placed on top of other sprites on the screen. That is, if sprite one and sprite two occupy the same spot on the screen, only sprite one will be visible.

To make the faces invisible, I could color each one black and place it on a black background. Then when the yellow spotlight- sprite was placed on top of a face (actually, on the bottom because it would be a lower priority), the face would show up against the spotlight.

As I said, this would work easily. The problem was, I needed more sprites. I wanted the apartment windows (black at first) to light up (then gradually fade) after they'd been zapped. Doing this using graphics on the bitmapped screen I was using looked tricky—and potentially slow.

SPRSAV to the rescue. I simply saved the faces as graphic shapes. Line 160 shows how. To save you from page flipping, here's what it looks like:

```
FOR T = 3 TO 8: SPRSAV T,TH$(T):NEXT
```

What is happening here is each time through the loop sprite number T's shape gets saved as a string variable, in this case TH$(T). In other words, now if I used the GSHAPE command to print TH$(3) to the screen, it would print a

Smoke-Busters demonstrates just some of the graphics capabilities tucked away inside every Commodore 128.

shape identical to sprite #3. Line 205 shows that happening.

The reason this solution worked in freeing up extra sprites for me is that my faces would never have to move once placed on the screen. Thus I could draw them using the built-in sprite editor, BSAVE them, BLOAD them back anytime, convert them from sprites to graphics strings, and then put them anywhere I wanted on the screen as many times as I wanted. A very integrated system, indeed.

So when I placed these graphic string faces on the screen, all I needed to do was color them black and make sure they were in the black windows. Line 200 and 205 did this.

Now I had freed up six sprites. These became my "apartment interiors." All they would be is rectangles which would sit blackly, invisibly on the screen, one beneath each face, until a face got zapped. At that point, by simply changing the proper sprite's color to white, it would appear that the apartment lights had come on.

How so? Because another capability of VIC is that it can, as an option, make sprites have a lower priority than any non-sprite objects printed to the screen. Since the faces and even the window frames were non-sprite graphics on the screen, they would show up against the now white sprite-retangle. You invoke this option by inserting a 1 in the fourth spot in the SPRITE command. See line 225 to see that happen.

Creating this retangular sprite shape was a loop away. Line 195 pokes in the right data to do make sprite one a retangle and line 205 saves this shape to sprites three through eight.

The sprite windows were now nothing more than the "light" in the apartments. By cycling through white—light gray-medium gray-dark gray—the lights could appear to be dimming. And that's just what I wanted so the gamester would know when time was running out.

The point here is, by transferring the sprite shapes to graphics via SPRSAV, I could, in effect, have as many stationary sprites as I wanted (with some limitations). And had I wanted to, I could have grabbed the shape back as a sprite and moved it again. In other words, now we can have as many sprites as we want, just as long as we only need to move eight at a time. In a future column, I'll create a jigsaw puzzle maker that relies on just that principle.

If you're confused at this point, take a break and type Smoke-Busters into your 128. Once you've seen and played the game, come back and re-read and things will make a lot more sense.

As always, save your work before you run it the first time. There are two machine-language routines in Smoke-Busters, and if you don't get your data exact, they could crash the program. Also, the first time you run the program, it will take over a minute to draw the screen (it will do this while your screen is blanked out so don't panic). Then it will save the screen to disk so that future loads will be much quicker. Be sure that the disk you save Smoke-Busters on has a least 65 blocks free.

Well, enough for openers. I hope I've whetted your appetite to see these sprite tricks in action. If not, read the following scenario. It ought to get you typing.

Oh, by the way—if you smoke, don't be offended by Smoke-Busters. It's all in fun. C

SMOKE-BUSTERS

It's a hot summer's night in the apartment. The year is 1936. All across town people are sitting at their windows, yearning for a wisp of a breeze. Sitting, and in some cases, smoking.

Alone in a drab flat, you sit at your window. You are not thinking about being hot. And certainly you are not smoking. What are you doing? Smoke-Busting!

How? With your Smoke-Out gun, naturally (patent pending). From your apartment window you aim the ray of sub-violet light across the alley. From window to window you scan, watching, waiting to catch a glimpse of a suddenly revealed cigarette. When you see someone light up, you light them up. Literally. Each time you fire on a smoking smoker, their apartment glows.

Ah, there's one now—it's Mugsy Potts (you always suspected he was a smoker). You fire. Zap! You got him. Mugsy's cigarette is blown away, his apartment lights up. But you cannot relish the victory for long—there are five more windows with five more neighbors with many more cigarettes to extinguish. And until you light up all six apartments, your task is incomplete.

So you continue panning the building, watching, zapping, smoke- busting. You zap another, and another, and you start to feel good. Three apartment windows now are aglow, their inhabitants exposed for their heinous acts. But the effects of the first blast are starting to fade. Mugsy's apartment is getting darker, and darker, and now black.

Nobody ever said it was going to be easy.

It gets worse. You accidently fire on Sweet Polly Peterson in apartment five. You lose precious time while she gives you a piece of her mind.

Two months of nightly smoke-busting later, you realize that this is not for the timid, nor the slow of reflex, nor the impatient. Not only that, you are beginning to realize that there's very little money in it, at least not in the year 1936. You realize that you are probably the only smoke-buster in the nation. You vow that someday there will an army of smoke-busters and you'll be the general.

So you write a manual on the subject. The manual gets rejected by several publishers all with the same response: "Get with it. This is 1936. Everybody smokes."

Fifty years later, your manual is discovered—by me—in an abandoned building on a wooden table next to a window, third story up. I write a computer program to simulate your nightly good deeds, following your descriptions to the tee.

I sell the program to *Commodore Magazine*. The editors ask, "What about the manual?" I say forget it. (This baby's mine, it may be worth something.) They say, fine, then we don't publish your program. I relent. Well, I relent a little. I summarize your manual into five smoke-busting tips.

1) Always keep your gun moving. If you go past a smoker, you still may have time to go back and get him.

2) Press your trigger quickly—don't hold it down. If you do, the cigarette will

disappear and the victim will become angry.

3) Practice your smoke-busting on the hottest of nights, when folks are sluggish. This means level 9.

4) Don't think that just because an apartment is lit up the inhabitant has given up smoking. He just might be sneaking another puff. Zap him again to re-fresh him (and buy more time).

5) Whatever you do, don't quit. Though the smokers smoke unpredictably, they all do smoke. If you've got five windows lit and Mr. Number Six just won't take a puff, keep zapping one through five until six is history.

The editors are satisfied and pay me well. As I return from the bank, I reflect on the fact that you did the gruntwork while I get the glory. You were simply a man before your time, I reason. My foot kicks an old cigarette package on the sidewalk. I stoop to pick it up and note the words on the side: "Warning—the Surgeon General . . ."

I straighten. I salute. Somewhere, I know, you are returning my gesture. ■

Before typing these programs, read "How to Enter Programs," and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

Smoke-Busters

```
10 COLOR 4,15:COLOR Ø,1
   :PRINT CHR$(14)"[CLEAR,SHFT SPACE,
   SHFT P]LACE JOYSTICK IN[SPACE2]
   PORT 2":SLEEP 2:PRINT"[CLEAR]";
   'GSOO
15 GOSUB 4Ø5:FAST'CFFE
2Ø GRAPHIC 1,1:COLOR 1,8:WIDTH 2'DKRC
25 DRAW 1,Ø,Ø TO 3,3:SSHAPE CIG$(1),Ø,
   Ø,23,2Ø'DBNK
3Ø DRAW 1,4,3 TO 7,Ø:SSHAPE CIG$(2),4,
   Ø,27,2Ø'DBFG
35 WIDTH 1'BCYF
4Ø CIRCLE 1,32,12,11,9:PAINT 1,32,12
   :SSHAPE ST$,21,1,44,21
   :COLOR 1,9'EOML
45 DOPEN#2,"SMOKESCREEN"
   :IF DS=Ø THEN DCLOSE
   :BLOAD"SMOKESCREEN":GOTO 14Ø'HPDT
5Ø DCLEAR:SCNCLR'CCCD
55 OF=-4:DRAW 1,8Ø,Ø TO 8Ø,199'EQOM
6Ø FOR T=Ø TO 199 STEP 5
   :FOR I= 84+OF TO 319 STEP 16
   :IF OF=4 AND I= 88 THEN BOX 1,8Ø,T,
   87,T+3,,1'QODV
65 BOX 1,I+1,T,I+15,T+3,,1:NEXT:OF=-OF
   :NEXT'IXIR
7Ø COLOR 1,15:BOX 1,16Ø,Ø,167,199,,1
   :BOX 1,Ø,96,319,1Ø3,,1'DQON
75 FOR T=1 TO 8:READ X:COLOR 1,6'FJVN
8Ø Y=24:IF T >4 THEN Y=12Ø'FKEJ
85 BOX Ø,X,Y-1,X+33,Y+15
   :BOX 1,X+1,Y,X+32,Y+14,,1
   :BOX Ø,X,Y+15,X+33,Y+27,,1
   :BOX Ø,X,Y+29,X+33,Y+56,,1'QUFL
9Ø DRAW 1,X+16,Y+15 TO X+16,Y+21
   :CIRCLE 1,X+16,Y+22,1,1'JFOS
95 NEXT'BAEK
1ØØ DATA 111,175,239,3Ø3,111,175,239,
    3Ø3'BGBB
1Ø5 FOR T=1 TO 2Ø:C=(T AND 7)+2
    :COLOR 1,C:RX=INT(RND(1)*78)
    :RY=INT(RND(1)*95)'PKCR
11Ø DRAW 1,RX,RY:NEXT'CIBY
115 COLOR 1,8:CIRCLE 1,18,15,24,14
    :PAINT 1,18,22:CIRCLE Ø,3Ø,18,22,
    12:PAINT Ø,3Ø,18'FWAO
12Ø COLOR 1,3:CIRCLE 1,4Ø,16Ø,3Ø,24
    :CIRCLE 1,4Ø,16Ø,24,2Ø
    :PAINT 1,14,16Ø'ERYI
125 COLOR 1,2:BOX 1,38,145,41,175,75,
    1'CYJI
13Ø COLOR 1,3:BOX 1,38,138,44,184,135,
    1'CAKE
135 BSAVE"SMOKESCREEN",BØ,
    P7168 TO P16384'CQPL
14Ø RESTORE 48Ø'BDOA
145 FOR T=2816 TO 2991:READ H$
    :POKE T,DEC(H$):NEXT'HUCM
15Ø FOR T=3584 TO 4Ø95:READ H$
    :POKE T,DEC(H$):NEXT'HUCI
155 SPRSAV 1,SM$:SPRSAV 2,CIG$(3)'CRQK
16Ø FOR T=3 TO 8:SPRSAV T,TH$(T)
    :NEXT'FOHH
165 SLOW:COLOR 1,2:CHAR 1,2,14,
    "FIRE TO":CHAR 1,3,15,"START"'EVKR
17Ø FOR T=1 TO 8:SP=INT(T/2)+4
    :SPRSAV 1,T:SPRITE T,1,T,Ø,1,
    T AND 1:MOVSPR T,19Ø,9Ø
    :MOVSPR T,T*4Ø#SP:NEXT'OBHX
175 GOSUB 425'BDOI
18Ø CHAR 1,2,14,"[SPACE7]"
    :CHAR 1,3,15,"[SPACE5]"'CPUJ
185 FOR T=1 TO 8:SPRITE T,Ø
    :MOVSPR T,Ø#Ø:NEXT'GQCP
19Ø GSHAPE SM$,3Ø,11Ø'BKSG
195 FOR T=3584 TO T+61 STEP 3
    :POKE T,255:POKE T+1,255
    :POKE T+2,Ø:NEXT'LDOW
2ØØ COLOR 1,1'BDQW
2Ø5 FOR T=3 TO 8:SPRSAV 1,T
    :READ X,Y,CIG(Ø,T),CIG(1,T)
    :GSHAPE TH$(T),X,Y,2
    :MOVSPR T,X+2Ø,Y+31'JEJT
21Ø NEXT:SPRSAV ST$,2'CHXY
215 FOR T=1 TO 8:READ X(T),Y(T)
    :NEXT'FOFI
22Ø E$=CHR$(27):J$=E$+"J":M$=E$+"M"
    :L$=E$+"L":SP$="[SPACE4Ø]"'JADP
225 SYS 2816:GRAPHIC 1:FOR T=3 TO 8
    :SPRITE T,1,1,1,1,1:NEXT'HYRM
23Ø SPRITE 2,1,8,1,1,1:MOVSPR 2,Ø#Ø
    :MOVSPR 2,128,96:SX=128:SY=96
    :S=1'GSSM
235 SPRSAV CIG$(1),1:SPRITE 1,1,2,Ø,Ø,
    Ø:MOVSPR 1,Ø#Ø'DFUM
24Ø SC$="":TI$="ØØØØØØ":POKE 252,Ø
    :G=99'ERTH
245 DO:G=G+1:IF G>LV*2+12 THEN
    BEGIN'JMUO
25Ø : S=INT(RND(1)*6)+3
    :IF S=S1 THEN S=S+2
```

```
    :IF S>8 THEN S=3'OVXP
255 : POKE 253,S:S1=S:G=Ø'DNQL
260 : CI=1:IF S=5 OR S=8 THEN
    CI=2'HMVJ
265 : BEND:SPRSAV CIG$(CI),1
    :MOVSPR 1,CIG(Ø,S),CIG(1,S)'DKLR
270 XY=JOY(2):IF XY<128 THEN 28Ø'FOKJ
275 GOSUB 3ØØ:IF PEEK(252)=1 THEN 345
    :ELSE 295'GRBP
28Ø SX=SX+X(XY):IF SX<128 OR SX>256
    THEN SX=SX-X(XY)'JENS
285 SY=SY+Y(XY):IF SY<96 OR SY>194
    THEN SY=SY-Y(XY)'JDEX
29Ø MOVSPR 2,SX,SY'BIDH
295 LOOP'BAKK
3ØØ XP=(SX-128)/64+1:YP=(SY-96)/96
    :JS=YP*3+XP+2:G=99'MLHO
3Ø5 IF JS<>S THEN 325'EGGG
31Ø SYS 29Ø7,Ø,Ø,JS-1'CMEB
315 IF JOY(2)>127 THEN 315'EJDH
32Ø RETURN'BAQY
325 POKE 253,JS'BGYF
33Ø SPRSAV CIG$(3),1:MOVSPR 1,CIG(Ø,
    JS)-4Ø,CIG(1,JS)-4Ø
    :SPRITE 1,1,2,Ø,1'FXON
335 SOUND 1,2ØØØ,5Ø,2,18ØØ,19ØØ'BVKJ
34Ø SLEEP 1:SPRSAV CIG$(CI),1
    :SPRITE 1,1,2,1,Ø,Ø:RETURN'EDRJ
345 SYS 65418:FOR T=1 TO 8:SPRITE T,Ø
    :NEXT'GPFN
35Ø GRAPHIC 2,Ø,13:PRINT"[CLEAR]"
    :FOR T=1 TO 12:PRINT:NEXT'HOJJ
355 PRINT"[RVS,L. BLUE]"SP$"[SPACE9,
    SHFT E]XTINQUISH TIME
    :[SPACE2]"MID$(TI$,3,2)"
    :"MID$(TI$,5)"[SPACE8]"'DTRV
36Ø RK=PEEK(161):RK=45-RK
    :IF RK<1 THEN RK=1'IVKN
365 IF RK>39 THEN RK=4Ø'EIIM
37Ø PRINT"[RVS]"SP$"[SPACE4,SHFT A]
    RROWS BELOW SCALE INDICATE RANK.
    [SPACE3]"'BDAP
375 PRINT"[RVS]"SP$"[GRAY3] [SHFT S]
    MOKERS'[L. BLUE,SPACE23,GRAY3]
    [SHFT S]URGEON"'BDAU
38Ø PRINT"[RVS,GRAY3] [SHFT F]RIEND
    [SPACE2]--[SPACE2,SHFT F]AIR
    [SPACE2]--[SPACE2,SHFT G]OOD
    [SPACE2]--[SPACE2,SHFT G]
    ENERAL"SP$"[UP]";'BEOS
385 FOR T=1 TO RK:PRINT"[UP ARROW]";
    :SOUND 1,2ØØ,4:NEXT:PRINT J$'HSWS
39Ø IF RK=4Ø THEN SOUND 2,12ØØØ,21Ø,2,
    1ØØØØ,5ØØ'ECLO
395 PRINT M$"[RVS,L. BLUE]"SP$"
    [SPACE8,SHFT P]RESS [WHITE]
    [SHFT P] [L. BLUE] TO PLAY AGAIN.
    [SPACE8]"SP$SP$J$"[DOWN3]";'BOHA
4ØØ GET KEY A$:IF A$="P"THEN GOSUB 4Ø5
    :GOSUB 435:GOTO 225:ELSE 4ØØ'JUPI
4Ø5 PRINT"[RVS,L. BLUE,SPACE3,SHFT C]
    HOOSE LEVEL:[SPACE2]
    1 (PRO) - 9 (BEGINNER)"'BALO
41Ø GET KEY A$:LV=VAL(A$)
    :IF LV<1 THEN LV=5'IQFI
415 POKE 251,(LV+5)*18:POKE 25Ø,Ø'ERGK
42Ø RETURN'BAQA
425 V$="V1TØO4IE$EEFE$EFSE"
    :Q$="MV2T6O2W":V3$="T3SO2C"'DJHR
43Ø FOR T=1 TO 6:READ N$
    :V$(T)=Q$+N$+V$:NEXT'ITOK
435 ENVELOPE 3,1Ø,Ø,3,Ø,3
    :ENVELOPE Ø, 4,4,Ø,Ø
    :ENVELOPE 6,8,3,3,Ø,2'DMXP
44Ø V$(7)="V2T6O2QE V1TØO4I EO3B
    V2O2Q#G V1O4IEF V2O2QB V1IEO3B
    V2O3QD V1IO4D#D"'BFWU
445 V$(8)="V1HO3A V2HO2A"'BFFM
45Ø TEMPO 2Ø:FOR T=1 TO 7
    :PLAY V3$+V$(T)+"M"'HRTL
455 IF JOY(2)>127 THEN RETURN'FGRM
46Ø NEXT:PLAY V$(8)'CHRG
465 IF RK=Ø THEN FOR T=1 TO 8
    :MOVSPR T,19Ø,9Ø:NEXT
    :GOTO 45Ø'JVXT
47Ø RETURN'BAQF
475 REM ML DATA'BGPM
48Ø DATA 78,A9,47,8D,14,Ø3,A9,ØB,8D,
    15,Ø3,58,6Ø,AØ,Ø2,2Ø'BWGP
485 DATA 55,ØB,C9,ØB,DØ,Ø8,A9,ØØ,99,
    27,DØ,4C,3F,ØB,C9,ØC'BWEV
49Ø DATA DØ,Ø8,A9,ØB,99,27,DØ,4C,3F,
    ØB,C9,ØF,DØ,Ø8,A9,ØC'BWYR
495 DATA 99,27,DØ,4C,3F,ØB,C9,Ø1,DØ,
    Ø5,A9,ØF,99,27,DØ,C8'BWFW
5ØØ DATA CØ,Ø8,DØ,CB,4C,71,ØB,2Ø,A2,
    ØB,C6,FA,DØ,F6,A5,FB'BWMJ
5Ø5 DATA 85,FA,4C,ØD,ØB,B9,27,DØ,29,
    ØF,6Ø,A9,Ø1,99,27,DØ'BWXO
51Ø DATA AØ,Ø2,2Ø,55,ØB,FØ,Ø9,C8,CØ,
    Ø8,DØ,F6,A9,Ø1,85,FC'BWEK
515 DATA 6Ø,AØ,Ø3,AD,Ø2,DØ,AE,Ø3,DØ,
    EØ,6Ø,FØ,Ø2,AØ,Ø6,C9'BWPP
52Ø DATA 8Ø,FØ,Ø6,C8,C9,CØ,FØ,Ø1,C8,
    A2,ØØ,C4,FD,DØ,Ø1,E8'BWKL
525 DATA 8E,FF,ØB,AD,15,DØ,29,FE,18,
    6D,FF,ØB,8D,15,DØ,4C'BWXQ
53Ø DATA 65,FA,CE,FE,ØB,DØ,Ø6,A9,Ø2,
    8D,FE,ØB,6Ø,E6,FA,6Ø'BWNM
535 REM SPRITE DATA'BKCK
54Ø DATA 3C,ØØ,ØØ,4Ø,ØØ,8Ø,42,ØØ,8E,
    73,ØC,AA,ØA,CA,AC,ØA'BWFN
545 DATA AA,C8,ØA,A6,AE,7Ø,AØ,9Ø,ØØ,
    2Ø,Ø8,CØ,1Ø,ØØ,AØ,3B'BWHS
55Ø DATA ØØ,A2,15,5B,AA,77,64,CA,94,
    47,AA,D7,41,96,58,11'BWDO
555 DATA 9Ø,4Ø,ØE,91,8Ø,ØØ,AØ,ØØ,ØØ,
    CØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ'BWKS
56Ø DATA Ø3,FF,EØ,1F,F7,BE,FB,FB,7F,
    BA,FD,7F,AA,EE,95,AA'BWJQ
565 DATA 2A,A5,A6,AA,C9,DE,B2,FF,FF,
    BB,FE,7F,E7,FC,ØF,FF'BWDV
57Ø DATA FØ,ØØ,FF,8Ø,ØØ,38,ØØ,ØØ,18,
    ØØ,ØØ,ØC,ØØ,ØØ,ØØ,ØØ'BWKP
575 DATA ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,
    ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ'BWXU
```

128 USERS/SMOKE-BUSTERS

```
580 DATA Ø1,F7,ØØ,Ø3,FF,8Ø,Ø3,FF,8Ø,
    E2,AA,8Ø,FF,FF,FØ,7F'BWKR
585 DATA FF,FE,1E,ØØ,47,3E,1F,FØ,78,
    FC,F8,F3,7C,38,FC,38'BWFW
59Ø DATA Ø8,EØ,ØØ,Ø4,E4,ØØ,Ø4,78,ØF,
    F8,3Ø,1F,F8,12,1F,F8'BWIS
595 DATA Ø9,ØØ,4Ø,Ø4,EØ,8Ø,Ø2,3F,ØØ,
    Ø2,Ø4,ØØ,Ø2,Ø8,ØØ,ØØ'BWLW
6ØØ DATA ØF,FF,CØ,9F,EF,FØ,EF,F7,F4,
    F4,7D,F8,BC,ØØ,1C,DC'BWYK
6Ø5 DATA ØØ,18,FC,ØØ,1C,BF,FF,F8,D7,
    DD,BØ,E3,F8,EØ,C8,ØØ'BWMP
61Ø DATA 3Ø,DØ,ØØ,Ø8,64,ØØ,84,38,ØØ,
    78,18,ØØ,1Ø,7C,Ø4,Ø8'BWYK
615 DATA FF,Ø8,7Ø,FF,C1,9Ø,17,FØ,1Ø,
    1D,FE,6Ø,3F,7F,8Ø,ØØ'BWLQ
62Ø DATA ØØ,Ø7,ØØ,ØØ,CF,ØØ,Ø1,FF,8Ø,
    Ø1,DF,8Ø,Ø1,FF,CØ,Ø3'BWQL
625 DATA FF,CØ,Ø3,7E,CF,Ø3,DB,FF,Ø7,
    FF,FF,1F,FF,EØ,3F,FC'BWKR
63Ø DATA FØ,79,ØØ,78,E1,ØØ,DC,82,1Ø,
    CC,Ø4,1E,4E,Ø4,3C,1E'BWDN
635 DATA Ø4,F8,1F,Ø3,EØ,3Ø,ØØ,2Ø,C8,
    ØØ,23,84,ØØ,3E,Ø2,ØØ'BWDR
64Ø DATA 1F,FE,ØØ,3F,FF,8Ø,78,ØØ,CØ,
    FØ,ØØ,6Ø,FF,FF,FØ,C7'BWHO
645 DATA FF,FC,FF,FF,FF,9Ø,FF,FF,9Ø,
    1C,1Ø,98,36,3Ø,AF,E3'BWCT
65Ø DATA EØ,6Ø,Ø2,2Ø,2Ø,1E,2Ø,1Ø,8Ø,
    4Ø,ØF,ØØ,4Ø,ØE,ØF,4Ø'BWPO
655 DATA ØB,1Ø,CØ,11,8Ø,4Ø,1Ø,EØ,8Ø,
    2Ø,3F,ØØ,2Ø,3Ø,ØØ,ØØ'BWLT
66Ø DATA ØØ,3F,ØØ,ØØ,7F,CØ,Ø3,FF,EØ,
    Ø7,FB,F8,ØF,F7,FC,3F'BWBQ
665 DATA FF,DE,7F,FF,FB,7B,FF,F7,7F,
    FØ,FE,77,EØ,FE,7F,EØ'BWDW
67Ø DATA 7C,7D,CØ,ØØ,FF,CØ,66,FF,8Ø,
    Ø8,DF,8Ø,Ø4,EF,2Ø,1C'BWRR
675 DATA FF,3Ø,ØØ,7F,EØ,78,7F,C8,ØØ,
    3F,87,1Ø,3F,Ø4,EØ,ØØ'BWAW
68Ø DATA 3D,FØ,ØØ,7F,FF,ØØ,7F,FE,CØ,
    3F,F3,F8,1Ø,Ø7,FC,1Ø'BWFS
685 DATA Ø7,FE,1Ø,Ø3,1E,ØE,63,4E,Ø5,
    58,4E,Ø2,2Ø,1C,Ø4,ØØ'BWYW
69Ø DATA 7C,Ø8,ØØ,38,Ø8,4Ø,18,Ø7,ØØ,
    1Ø,ØØ,8Ø,1Ø,ØØ,71,2Ø'BWPS
695 DATA ØØ,42,2Ø,ØØ,44,2Ø,ØØ,38,1Ø,
    ØØ,Ø8,1Ø,ØØ,Ø8,Ø8,ØØ'BWCX
7ØØ DATA A,$A,G,$G,F,E'BNPE
7Ø5 DATA 116,58,158,124,18Ø,58,224,
    125,244,58,272,127,116,154,158,
    221,18Ø,154,224,222,244,154,272,
    22Ø'BRAY
71Ø DATA Ø,-96,64,-96,64,Ø,64,96,Ø,96,
    -64,96,-64,Ø,-64,-96'BYAL
```

END

Interfacing Commodore's User Port

Continued from pg. 80

```
30 D=PEEK(56577)AND 128
40 IF D=0 GOTO 30
50 POKE 56577,X
60 FOR T=1 TO 100:NEXT T
70 NEXT X
80 GOTO 10
```

To fully understand this we must look at some logic instructions. You may skip this section if you feel it's too difficult, but this logic section will enrich your ability to design and configure the user port to your needs with the minimum amount of instructions.

Logic

There are two logic instructions available to us from BASIC. That can be used to set (binary 1) or clear (binary 0) specific bits on the port without affecting the other bits. They are AND and OR instructions.

When using these instructions, we are comparing the number in the register to the number we peek or poke in the register. The results can be used to make useful decisions and perform functions. For each set of bits compared there are four possible combinations.

0	0
0	1
1	0
1	1

Here are the results of these two instructions.

	AND	RESULT		OR	RESULT
0		0= 0	0		0= 0
0		1= 0	0		1= 1
1		0= 0	1		0= 1
1		1= 1	1		1= 1

By studying these results, two conclusions can be drawn. The results of an AND instruction is 1 only if both bits are 1, otherwise the results are 0. And the results of an OR instruction is 0 only if both bits are 0, otherwise the results are 1.

Our computer uses eight-bit binary numbers. Examine the following examples.

AND		OR	
11010011	Register (211)	11010011	Register (211)
10000101	AND 133	10000101	OR 133
10000001	RESULT (129)	11010111	RESULT (215)

In the fourth program we used the AND instruction to test a bit. Then we made a decision based upon the results. Let's analyze how the program accomplishes this.

30 D=Peek	(56577)AND128	
0XXXXXXX	Register	X=any Value 0 or 1
10000000	AND 128	
00000000	RESULT	PEEK(56577)AND128=0

This instruction compares the AND 128 with the number in the register. The only bit that can have an impact on the result is bit 7. Because all other bits are AND with 0, their results are 0. By setting bit 7, we have the following scenerio.

1XXXXXXX	Register X = any value 0 or 1	
10000000	AND 128	
10000000	RESULT PEEK(56577)AND128=128	

With these two possible results, we can use a familiar basic decision command 40 IF D=0 THEN GOTO 30.

Setting a Bit With OR

We can use an OR instruction to set various bits. It is very useful when we wish to set specific bits without disturbing the status of the other bits on the port. Any number that is OR with a 0 remains unchanged. Therefore, if we wish to set bit 4, we can OR bit 4 with a binary 1 as our example illustrates.

10100010	Register	
00001000	OR 8	Command=POKE56577,PEEK
10101010	RESULT	(56577)OR8

Try entering the following commands to get a better understanding.

POKE 56579,15	DDR Set-up.
POKE 56577,3	Lights LEDs to PB0 and PB1.
POKE 56577,PEEK(56577)OR8	Sets bit 4 on without disturbing the status of the other bits.

When we enter our last command, the status on the interface has the two LEDs lit. This status remains unchanged as we set bit 4 on, as indicated by the lit LEDs.

There are other interesting bit manipulations, but we do not have the time to explore further. For those interested in becoming more proficient, I advise you to experiment on your own and pick up a book on machine-language programming. ■

BY MATT BLAIS

Curing the Red-Green Blues



ART BAXTER

"Aaaargh!" I cried. "There must be a better way to make colors than this. You wouldn't give Michelangelo three cans of paint—red, green and blue—and ask him to mix all the colors of the rainbow from those, would you?"

A voice spoke from my shoulder, "Gee, that's a wild shade of pink. What are you drawing?"

"A portrait," I mumbled. "That's supposed to be flesh-color, not punk-neon pink. This Amiga is an incredible machine—4,096 colors, no less—but whoever heard of creating a palette from three colors? I'm having the devil's own time trying to get shades and hues this way, mixing red, green and blue."

"I know someone else who has an Amiga. Maybe he can help?"

"You don't mean...?"

"None other. The Wiz has arrived on the Amiga scene and Amigaland will never be the same. He must have sold that baseball card collection."

More likely he's in hock up to his eyeballs, I thought. Now, I'm not one to go begging for favors, but I was at the end of my artistic rope trying to turn red, green and blue into a robust Floridian tan. Besides, it had been ages since I'd paid the Wiz a visit. The last time, he'd had a Commodore 64 system on which he would perform miraculous feats of graphics and sound programming. I was eager to see what the Wiz unleashed could do with an Amiga, the ultimate graphics and sound machine.

"So what are we waiting for?"

The Wiz sat perched in front of his Amiga, intent on the stream of characters flowing by on the screen. He seemed not to notice us standing just behind him.

"Shh—debugging!" he blurted, as I was about to tap his shoulder. I snatched my hand back. One does not disturb the Wiz in debug-mode, unless one enjoys experiencing the full unbounded fury of a fourteen year-old.

"Aha! There's the little bugger!" he exclaimed. "Thought you could hide from me in there, did you? He twirled around in his chair to face us, "What can I do for you gentlemen?"

"He's got a problem making colors on his Amiga," volunteered my friend. I stepped on his toes and glared, and he elbowed me and glared back, but shut up.

"Well actually, it's not really a problem... or maybe just a little problem. You see, I'm using this graphics program to draw nude—er, still-lifes, and I know the Amiga can give me all the colors I need, only flesh isn't red, green or blue, and I can't quite seem to get the colors I want, but what I really need. . ."

"What you really need is some way to generate colors other than by mixing red, green and blue," said the Wiz, finishing my sentence.

"Yeah, I want to be able to pick a color and look at its shades and tints, and not have to try and figure out how to mix them from red, green and blue myself."

The Wiz creased his brow and looked thoughtful for a minute. When he does that you can almost hear little cogs and disks whirring and clicking away behind those coke bottle bottom glasses of his. Then, with an aura of profound sagacity, he intoned, "Hue, Saturation, Luminance. C'est ca."

My friend and I exchanged puzzled glances. The Wiz explained, "That's French for 'That's it.' You need a program that lets you choose colors by hue, saturation and luminance instead of red, green and blue."

"Of course," said my friend. I could tell he was faking. It's a bad habit of his.

I didn't feel enlightened by the explanation, and it must have showed, as the Wiz continued: "Every color has a hue value, a saturation value and a luminance value. You can make any color by mixing red, green and blue light because those are the primary additive colors. . ."

"That's how television works," interrupted by friend. Suddenly he was oozing knowledge.

Ignoring the interruption, the Wiz continued, "But you can also specify a color by its hue, saturation and luminance. Imagine a color wheel where you have red on the right at zero degrees, green at the upper left at 120 degrees, and blue at the lower left at 240 degrees, with all the different shades in between. Any particular color in the spectrum is found somewhere on the color wheel, and its position there in degrees is its hue value."

"Roy G. Biv." remarked my friend cryptically. "Now I remember."

"I don't," I said foolishly.

"Didn't they teach you anything in science class?" he quipped. "Red, Orange, Yellow, Green, Blue, Indigo, Violet: Roy G. Biv."

"It's an acronym for the colors in the spectrum in the order that they occur," explained the Wiz.

"Of course," I replied (it was my turn to fake it), "but that's only seven colors—where's brown? And what about fleshtones?" I always was suspicious of science teachers. Where had they hidden

brown, anyway?

"That's where saturation and luminance come in," the Wiz continued. "Ever notice how those seven hues on the color chart—and even the shades in between—are all bright? Ever wonder what happened to all the dull colors, like the browns, grays and beiges?"

"Yes, yes, and the fleshtones," I said impatiently. "So if they're not on the color wheel, where are they?" "Well," he continued leisurely, "all of those colors on the wheel are what we call saturated. That is, the red on the color wheel is as red as red can get, and the blue is as blue, and so on. The colors which aren't on the color wheel are the less-saturated colors: less-saturated red is pink, less-saturated blue is light blue. You can make any color lighter by reducing its saturation."

"Okay," I countered, "so where's brown? That's not lighter- anything, except lighter dark-brown, which isn't on the wheel. And what about darker blue?" I was sure I had him now.

"I was just getting to that," he said. "That's where luminance comes in." Oops, I'd forgotten about luminance. "By decreasing saturation, you can make any color lighter, until it becomes white, but by decreasing luminance, you can make any color darker, until you get black. Brown is really just dark orange, orange with a low luminance. Flesh-tones are light reds and oranges, darkened a little."

"Now wait just one minute here," interjected my friend. He hadn't interrupted for so long I'd forgotten he was there. "How can you have a darkened light color? Doesn't darkening just undo what lightening did?"

"You're mistaking lighten for brighten," explained the Wiz.

"Brighten is the opposite of darken. When you lighten a color, it doesn't necessarily get any brighter—it just fades toward white. It's like watering down the color, making it less colorful. Red fades through light red, then pink, then a pinkish tint of white, and finally to just white. That's what decreasing a color's saturation does."

"So what do you get," asked my friend, "if you take a light pink and then darken it?" Good questions, finally, I thought. Maybe there's hope for him yet.

"It still stays a light pink color, but it gets dimmer, as if you were turning down the light. That's what the luminance of a color does. With full luminance, a color is as bright as it can get, and as you decrease the luminance, it just gets dimmer and dimmer until there's nothing left at all, nothing but black."

The Wiz turned then and started flipping through a long box of 3.5-inch Amiga diskettes, all the while muttering to himself: "...one of the first programs I wrote... when I got this thing... here we are!" He blew a thin layer of dust off of a disk and popped it in his Amiga. "Here's a little AmigaBasic program I whipped off in a few minutes that should help you out nicely."

I glanced at the screen as the program scrolled endlessly by. "You say little?"

"Oh, it's mostly comments. The real meat of it—the part that translates hue, saturation and luminance into red, green and blue—is only eight statements long. The rest of it just draws fancy pictures to let you see your palette. You can make a palette with up to 32 colors, which is as many as the Amiga will display at once, normally." He ran the program and a palette of beautiful colors splashed across the screen in bars and circles.

"Pretty, huh? That's just the beginning. You can look at every shade of every hue here. These are only 24 of them—you still have 4,072 more to go.

"What this program lets you do is create a palette by specifying only the colors at its two ends. The program fills in the colors in the middle so you have a smooth transition from one end to the other.

"You select each end color by specifying its hue, saturation and luminance values, not its amounts of red, green and blue. The variables h, s and l are the hue, saturation and luminance values of the first color in the palette, and h2, s2 and l2 are the corresponding values for the last color. You can choose the number of colors you want in the palette by setting the variable nc.

"Also, if you don't like the gray and black colors of the screen, you can change them by changing the PALETTE statements in the main paragraph. But remember, the PALETTE command needs red, green and blue arguments, not hue, saturation and luminance."

My head was beginning to spin. "So... if I want to see some fleshtones?" I asked weakly.

"We'll start with violet, which has a hue value of 330, for h, and go through a little yellow—say, to 40 degrees—for h2. Then we'll want a fairly low saturation to lighten it up—about 0.5—for s and s2; and we don't want it too bright, so we'll try a luminance of 0.9 for l and l2.

Now, this is only changing hues, because we've specified the saturation and luminance as the same for the first and last color. If you wanted to see all the bright reds and pinks, for instance—from red all the way through white—you would make the starting and ending hues both zero, but the starting saturation at 1.0, which is pure red, and the ending saturation at zero, which is white, and both luminances at 1.0. Play around with it and see what you get."

And so, I took the Wiz's advice and typed in his program (he was too cheap to give it to me on a disk). And oh, what marvelous colors I now have: tints and shades and hues galore! I found that robust Floridian tan in no time, and discovered some tantalizing Mediterranean-bronze hues hiding in there as well. The Hues program is civilized enough to tell you what the red, green and blue values of the entire palette were—or even write them to a file on disk— after you've finished ogling it. Then if you want to use the palette or a part of it in another program, you just use the numbers. They're in hexadecimal (base 16) and there's a routine at the end of the Hues program which will read them in and convert them to unit values for the AmigaBasic PALETTE statement.

One other thing: Because there are only 4,096 possible colors, if you use 30 colors for a palette which changes very little from beginning to end, you'll get a lot of identical colors next to each other in your palette, or even a few colors which are a little off because of rounding errors. These palettes will be vastly improved if you use fewer colors, say, 20 or 14.

If you have a graphics program like mine that lets you use sliders to set the red, green and blue amounts for colors, and you want to use colors generated by Hues, first note the red, green and blue values for each of the colors you want to use from the list that is generated after the palette has been displayed. Then when you are setting the colors in your graphics program, start each slider out at zero and click the pointer just to the right of the slider's marker once for each count. Each click should advance the slider by one- sixteenth of its total range (or one count).

For example, if you want to set a color

which has a red value of 3, a green value of 0, and a blue value of 10, first set all three sliders to zero, then click beside the red slider three times, then beside the blue slider ten times, leaving the green slider at zero.

Oh by the way, I should have warned you—the Wiz is prone to fits of fantastic exaggeration. The program's not really mostly comments any more than it took him a few minutes to whip it off.

Hues

```
REM ************************************************
REM **                                            **
REM **   Program:   Hues                          **
REM **   Author:    Matt Blais                    **
REM **   Date:      18-Jun-1986                   **
REM **                                            **
REM **   This program generates and displays a    **
REM **   palette using a HUE-SATURATION-LUMINANCE **
REM **   algorithm (converting HSL to RGB).       **
REM **                                            **
REM **   The palette is defined by the HSL values **
REM **   of its first and last color, and the colors **
REM **   in between are interpolated by the program. **
REM **   See the first few lines in 'MakePalette' to **
REM **   change the first/last color HSL values.  **
REM **                                            **
REM **   After displaying the palette, the program **
REM **   dumps its RGB values (in the form of DATA **
REM **   statement(s) to the screen or to a file for **
REM **   use in your programs.                    **
REM **                                            **
REM ************************************************

Hues:
  DEFINT a-z
  DEFSNG r,g,b,h,s,l,m
  DIM red(32),grn(32),blu(32),pat%(1)

  DEF FNi(r) = INT(r*16+.5)+(r>.96875)
                                  '— PALETTE value to INT.
  DEF FNh$(i) = MID$("0123456789ABCDEF",i+1,1)
                                  '— INT to Hex char.

  pi! = 3.141593
  xmax = 311 : ymax = 186 '— Window dimensions.

  SCREEN 2,320,200,5,1
  WINDOW 2,"Hit Any Key To Continue",(0,10)-(xmax,ymax),0,2
  WINDOW OUTPUT 2
  PALETTE 0, .5,.5,.5
  PALETTE 1, .5,.5,.5             '-> Foreground color.

  GOSUB MakePalette
  ' GOSUB ReadPalette             '— You could do this instead
                                  ' of MakePalette
  PALETTE 0, 0, 0, 0              '-> Background color.
  GOSUB DrawBoxes
  GOSUB DrawWheel1
  GOSUB DrawWheel2

Finish:
  WHILE INKEY$ <> ""              '— Throw away all buffered
                                  ' keypresses
  WEND
  WHILE INKEY$ = ""               '— Wait for the next key.
  WEND

  WINDOW CLOSE 2
  SCREEN CLOSE 2
  GOSUB ListPalette               '— List palette data to a file
                                  ' or to screen.
END

MakePalette:
  '— Hue is specified in degrees (from 0 to 360). Pure Red is
  '— at 0, Yellow at 60, Green at 120, and Blue at 240, with the
  '— various shades somewhere in between. Values greater than
  '— 360 wrap back around thru zerom, so you can go from 240 to
  '— 420 (60), for example. Saturation ranges from 0 to 1,
  '— where 1 is full color, which fades to gray as saturation
  '— falls to zero. Notice that if saturation is zero, it makes
  '— no difference what value hue has: it's all gray. Lumin-
  '— ance also ranges between 0 and 1; 1 gives you full bright-
  '— ness, which gets darker as luminance falls to zero (black).
  '— If luminance is zero, you get black regardless of hue and
  '— saturation.
  h = 0                  '-> First color's Hue.
  s = .8                 '-> First color's Saturation.
  l = .9                 '-> First color's Luminance.

  h2 = 348               '-> Last color's Hue.
  s2 = .8                '-> Last color's Saturation.
  l2 = .9                '-> Last color's Luminance.

  nc = 24                '-> Number of colors in this palette.
  IF nc>32 OR nc<3 THEN
    WINDOW CLOSE 2
    SCREEN CLOSE 2
    PRINT : PRINT "'nc' must be between 3 and 32!"
    END
  END IF
  ncl = nc - 1
  fc = 32-nc             '— ID of first color in palette.
  h3 = (h2-h)/ncl        '— Hue increment.
  s3 = (s2-s)/ncl        '— Saturation increment.
  l3 = (l2-1)/ncl        '— Luminance increment.

  FOR i=fc TO fc+ncl     '— For each palette color:
    GOSUB MakeRGB        '— Convert HSL to RGB.
    PALETTE i, r,g,b     '— Set palette.
    h = h + h3           '— Increment Hue.
    s = s + s3           '— Increment Saturation.
    l = l — l3           '— Increment Luminance.
    red(i-fc) = r
    grn(i-fc) = g        '— Save r, g, and b values for later.
    blu(i-fc) = b
  NEXT
RETURN

MakeRGB:        '— Converts h,s,l to r,g,b.

  r = (120-ABS((h+180)MOD 360-180))/120
  g = (120-ABS((h+ 60)MOD 360-180))/120
  b = (120-ABS((h+300)MOD 360-180))/120
  m = -ABS(h MOD 120-60)/120-.5
  sl = l-s
  r = (r/m*(r>0)*s+s1)*1
  g = (g/m*(g>0)*s+s1)*1
  b = (b/m*(b>0)*s+s1)*1
RETURN

DrawBoxes:
  xw = (xmax-16)/nc/2    '— Width of each box.
```

```
  y = 2                          ' — Y-position of each box.
  yh = 42                        ' — Box height.
  x1 = (xmax-xw*nc*2)/2          ' — X-position of first box.
  x = x1

  FOR i=fc TO fc+ncl
   LINE (x,y)-STEP(xw,yh),i,bf
   x = x + xw
  NEXT
  FOR i=fc+ncl TO fc STEP -1     ' — Now go backwards thru palette.
   LINE (x,y)-STEP(xw,yh),i,bf
   x = x + xw
  NEXT

  y = y+yh/2
  LINE (x1,y-2)-(x,y+2),0,bf     ' — Draw horizontal bar.
  x = x1 + xw
  FOR i=1 TO nc*2                ' — Draw vertical bars.
    LINE (x,y+1)-STEP(0,yh/2),0
   x = x + xw
  NEXT
RETURN

DrawWheel:
  pat%(0) = &HFFFF               ' — Fill pattern for wheels (solid).
  pat%(1) = &HFFFF
  PATTERN ,pat%
  xr = 70                        ' — X-radius of wheels.
  yr = .88*xr                    ' — Y-radius.
  x = xmax/4                     ' — Wheel #1's X-center.
  y = ymax-yr-13                 ' — Y-center.
  rn = 0                         ' — Initial position (radians).
  rs = 2*pi!/nc                  ' — Wedge step (radians).
  FOR i=0 TO ncl
    COLOR fc+i
    GOSUB DrawWedge              ' — Draw wedge at rn radians.
   rn = rn + rs                  ' — Increment around the circle.
  NEXT
RETURN

DrawWheel2:
  rs = rs/2                      ' — These wedges are 1/2-size.
  x = xmax*3/4                   ' — Wheel X-center.
  rn1 = 0                        ' — Top half position (radians).
  rn2 = 2*pi!-rs                 ' — Bottom half position (radians).

  FOR i=0 TO ncl
   COLOR fc+i
   rn = rn1
   GOSUB DrawWedge               ' — Draw top wedge.
   rn = rn2
   GOSUB DrawWedge               ' — Draw bottom wedge.
   rn1 = rn1+rs                  ' — Increment around 1/2 a circle.
   rn2 = rn2-rs
  NEXT
RETURN

DrawWedge:     ' — Draw and fill one triangular sector of a circle.
  AREA (x,y)
  AREA (x+xr*COS(rn), y-yr*SIN(rn))
  AREA (x+xr*COS(rn+rs), y-yr*SIN(rn+rs))
  AREAFILL
RETURN

ListPalette:
  ' — List the red, green, and blue values of
  ' —      each color in the palette.

  PRINT : PRINT "Enter file name to save palette data to disk"
  INPUT "or <RETURN> to list data to screen "; f1$
  IF f1$="" THEN
    f1$ = "SCRN:"
  ELSE
    PRINT: PRINT "Type in a SHORT one-line description of this palette:"
    INPUT desc$
  END IF
  PRINT
  IF desc$="" THEN desc$ = "Number of colors in this palette."
  OPEN f1$ FOR OUTPUT AS #1
  p1 = 8                         ' — # of r,g,b sets per data line.
  i = 0                          ' — I is current color # in r%(), etc.
  PRINT #1, "DATA"; NC; ": ' — "; desc$

  FOR j=1 TO INT(nc/pl+.99)
    PRINT #1, "DATA ";
    c i+pl-1                     ' — C is last set # on current data
    IF c>ncl THEN c = ncl ' — line.
    FOR k=i TO c
    r$ =
    FNh$(FNi(red(i)))            ' — Convert INTs to hex characters.
    g$ = FNh$(FNi(grn(i)))
    b$ = FNh$(FNi(blu(i)))
    PRINT #1, r$ + "," + g$ + "," + b$;
    IF k<c THEN PRINT #1, ", ";
    i = i + 1
   NEXT
   PRINT #1, ""
  NEXT
  CLOSE #1
  PRINT
RETURN

ReadPalette:
  ' — You can use this routine in your own programs to read in
  ' — palette data. To get the DATA statements from a file into
  ' — your AmigaBasic program, load your program and type "merge
  ' — XX", (XX is the data-file name) and the DATA statements
  ' — will be at the end of your program. FNd(x$) converts and
  ' — scales a hexadecimal digit to 0 - 1.

  DEF FNd(x$) = ASC(UCASE$(x$))-48+(x$>"9")*7   ' — Hex
                                                ' digit char.
                                                ' to INT.
  RESTORE Palette1
  READ nc                        ' — First data item is # of colors in palette
  ncl = nc-1
  fc = 32-nc                     ' — Fc is the ID of the first color.
  FOR i = 0 TO ncl
    READ r$,g$,b$
    r = FNd(r$)/16               ' — Convert hex digits ("0" - "F")
    g = FNd(g$)/16               ' — into unit values (0 - 1) for
    b = FNd(b$)/16               ' — PALETTE statement.
    PALETTE fc+i, r,g,b
    red(i) = r                   ' — Save palette r, g, and b values for later
    grn(i) = g
    blu(i) = b
  NEXT
RETURN

Palette1:
  DATA 15 : ' — Violet -> Yellow palette.
  DATA E,3,E, E,3,C, E,3,A, E,3,8, E,3,6, E,3,5, E,3,4, E,3,3
  DATA E,4,3, E,5,3, E,6,3, E,8,3, E,A,3, E,C,3, E,E,3
```

END

BY JOHN J. KOTTLER

Amiga Date-Setter

Keep your files straight by tracking when you did what.

Did you ever check the directories on your Amiga disks? Of course you have. But did you ever use the LIST command from the CLI? Well, try it now and chances are you will find plenty of your programs dated by "FUTURE" instead of another date. If you do update the date every time you use the computer, doesn't it take quite a while to load and use the Preferences program? If you really want to keep your files straight and keep track of when you did what, here is an easier way.

Command Line Interpreter

The Command Line Interpreter (CLI for short) icon lets a user use the AmigaDOS operating system inside of the Amiga. AmigaDOS is quite similar to MS-DOS or other IBM DOS systems. Once you have entered CLI, you have access to many disk commands. You can format disks, copy single files, and copy an entire disk, to name a few.

The CLI includes two commands that list the disk directory, DIR and LIST. DIR lists the disk's contents to the screen in alphabetical order with subdirectories at the top. LIST lists the contents along with the dates they were saved. If you do not update the date and time, you may find words like "FUTURE" replacing normal dates. To avoid this problem, you can update the date with the Preferences program. However, this is time-consuming and often impractical.

Start-up Sequence

The answer lies in a small file known as the Start-up Sequence, which is found under the S directory. This Start-up Sequence file is the file that prints the message on the screen when the Amiga boots. The trick is to alter this file.

The first step is to enter the CLI mode if you are not already in it. There are two ways to enter the CLI mode. First, if your Amiga is already booted, you can select the CLI icon from the Utilities drawer on the Workbench Disk, or second, if you insert the Workbench Disk after you have reset the computer, you can hold CONTROL-D to stop the Amiga after the blue screen is displayed. In either case, the prompt "1>" will be displayed inside the window named AmigaDOS. Now you can use the Amiga's disk commands.

I suggest that you first make a copy of your Workbench onto a blank disk so that you will not destroy the original. This can be accomplished by typing DISKCOPY DF0: TO DF1: if you have more than one drive. Change the DF1: to DF0: if you have only one drive.

After you have typed this, you will be reminded to insert the original disk into the internal drive and blank disk into the external drive #1. You should have the write-protect notch in the read-only position on your original so you will not erase it by mistake. When you're ready, press RETURN and the process will begin.

The current track number is always displayed so you know how much longer you have to wait. You should expect to see 79 when you are finished (0-79 is 80 tracks when track 0 is included). Don't remove the disks right away, because the Amiga needs a second to initialize both of the drive units. Now that you have a second copy of Workbench, you do not have to worry about changing the original Start-up Sequence.

Using the ED Command

The ED command is a complex command that edits text files. Even though the Start-up Sequence file is a boot-file, it can still be edited like a text file. To edit Start-up Sequence, enter the following line, assuming that your copy of Workbench is now in the internal drive unit: ED "DF0:S/STARTUP-SEQUENCE".

After a second, the screen will display the contents of the file. The file should contain several ECHO commands, a LOADWB command and an ENDCLI command. The editor works similarly to the new AmigaBASIC. Use the arrow keys to move around the screen and use the delete and backspace keys to erase the text from both directions. Each time you type a character, it will be inserted instead of written over. Once you become acquainted with the text manipulation, you can edit the Start-up Sequence file.

Echo . . . Echo . . . Echo . . .

No, the ECHO command doesn't actually echo—it acts like the BASIC PRINT statement. Just as you would type PRINT "I LOVE MY AMIGA!" you can type ECHO "I LOVE MY AMIGA!".

First, delete the ECHO that says "Use Preferences tool to set date." Wipe it out completely. Now on that same line, type the following with a return after each line.

```
ECHO "Hello! The LAST date saved was:"
DATE VER DATEFILE
ECHO " "
ECHO "What day is it today? (DD-MMM-YY)"
DATE ?
ECHO " "
ECHO "What time is it? (HH:MM:SS)"
DATE ?
ECHO " "
DATE TO DATEFILE
ECHO "Today's date is:"
DATE
```

Now to explain what is happening. First, the message "Hello! The last date saved was:" appears followed by whatever date the Amiga can remember. Obviously, when the computer asks what day and time it is, you can enter it after the prompt is displayed. However, you may have noticed that there are two "DATE?" commands to set both the date and time. The only way the Amiga can distinguish between a date entry and a time entry is by the way it was entered. When asked to enter the date, you should use the "DD-MMM-YY" method. For example, to enter "December 25, 1985" you would enter "25-Dec-85." Each "MMM" stands for the first three letters of the month.

Once you are prompted to enter the time, you can use either the HH:MM:SS or the HH:MM method. For example, 2 o'clock PM would be entered as military time 14:00. If you wanted to change it to quarter after two in the afternoon plus 36 seconds, you would enter that as 14:15:36.

After you have entered the correct date and time, the Amiga will resave the

Continued on pg. 127

BY TIM JONES

AmigaBASIC Tutorial

Part 3: Mouse

A short program demonstrates difficult areas of AmigaBASIC, offering both tips on confusing areas and tricks to make your life easier. Rather than an article followed by a program, we have combined them for the sake of clarity.

```
'
' This file contains information pertaining to the use of the
' MOUSE(X) function. This is what is used to monitor the left
' MOUSE button. It doesn't effect the right, or MENU, button
' at all.
'
' MOUSE(X) parameters can be numbers from 0 to 6 with each
' pertaining to a specific function. These functions are as
' follows:
'
'   0 ->  Keeps track of the status of the left mouse button.
'         It will return one of the following values:
'
'         0 —>>  The left button is not currently down.
'         1 to 3 -> The button is not currently down, but has
'                   been clicked the # returned times since
'                   the last call to MOUSE(0).
'        -1 to -3-> The button is being held down after being
'                   clicked the # returned times. These
'                   values usually indicate that the mouse is
'                   being moved.
'
'   1 ->  Returns the current X coordinate. This is the
'         coordinate at which the pointer was located when the
'         MOUSE(0) call was made, irregardless of whether the
'         button is being held down.
'
'   2 ->  Returns the current Y coordinate. This is handled
'         in the same manner as MOUSE(1) above.
'
'   3 ->  Returns the Starting X coordinate of the pointer
'         from the last time that the button was pressed
'         before MOUSE(0) was called. It is used in
'         conjunction with MOUSE(4) to get the starting X and
'         Y coordinates for things like line, box and circle
'         work.
'
'   4 ->  Returns the Starting Y coordinate of the pointer
'         from the last time that the button was pressed
'         before MOUSE(0) was called.
'
'   5 ->  Returns the Ending X coordinate of the pointer if
'         the button was being held down the last time that
'         MOUSE(0) was called. If the button was not held
'         when the last MOUSE(0) was called, it will return
'         the X position at which the button was released.
'         This, in conjunction with MOUSE(6), will allow you
'         to keep track of pointer movement and determine the
'         X and Y coordinates where movement stops.
'   6 ->  Returns the Ending Y coordinate in the same manner
'         as MOUSE(5) above.
Title$ = " Mouse(X) Tutorial                          Tim Jones
ScreenDef:
' We'll just use the standard WORKBENCH screen for this one.
WindowDef:
WINDOW 2,Title$,,8
PALETTE 1,1,1,1:PALETTE 0,0,0,0:PALETTE 3,.9,.9,0:PALETTE 2,0,.6,.3
  Ja = WINDOW(7)
Info:
  CLS : LOCATE 2,1 : COLOR 2,0
  PRINT "      Mouse(X) demonstration. As you operate the"
  PRINT "      Mouse, the different values will change on"
  PRINT "      screen. Please keep in mind that there is"
  PRINT "      quite a bit of updating going on, so forgive"
  PRINT "      me if this demo seems sluggish. You most"
  PRINT "      likely won't use ALL of these ALL of the"
  PRINT "      time."
  PRINT : PRINT "      Move the Mouse and Click, or  Click and "
  PRINT " Hold, the left button to see the changes."
  LOCATE 18,1 : PRINT "      Notice how MOUSE(5) and MOUSE(6)"
  PRINT "      change when you hold do button. These"
  PRINT "      registers are the most important for keeping"
  PRINT "      track of location for things like drawing"
  PRINT "      and targetting (in games)."
  PRINT : PRINT " Click on the CLOSE gadget to end this demo."
MainLoop:
  WHILE WINDOW(7) = Ja
    MOUSE ON : ON MOUSE GOSUB MouseLoop
    A3 = 0 : A4 = A3 : PALETTE 3,RND,.3,RND
    FOR Delay = 1 TO 200 : NEXT Delay
    GOTO MainLoop
  WEND
  WINDOW CLOSE 2 : WINDOW OUTPUT 1
  STOP
MouseLoop:
  zero = MOUSE(0) ' These values are held as variable
  one = MOUSE(1) ' to preserve them for our use in case we
  two = MOUSE(2) ' don't respond to the button right away.
  ' The others are to be handled as they occur and not as
  ' variables. This will allow for more instantaneous
  ' control over the results of those registers.
COLOR 1,0
LOCATE 10,34 : PRINT "Mouse (0) = ",zero,"          "
PRINT
PRINT "       Mouse(1) =";one,"Mouse(2) = ",two"      "
PRINT
PRINT "       Mouse(3) =";MOUSE(3);" Mouse(4) =";MOUSE(4);"      "
WHILE MOUSE(0) < 0
  COLOR 1,0
  LOCATE 16,24
  PRINT "Mouse(5) =";: COLOR 3,0 : PRINT MOUSE(5);:
  COLOR 1,0
  PRINT " Mouse(6) =";: COLOR 3,0 : PRINT MOUSE(6);"    "
  A1 = MOUSE(5)
  A2 =    MOUSE(6) : PSET (A1,A2),3
  IF A3 <> 0 OR A4 <> 0 THEN
    LINE (A1,A2) - (A3,A4),3 ' PSET and LINE will be covered
  END IF                     ' in parts 4 and 5 of this
                             ' series
  PALETTE 3,RND,RND,RND
  A3 = A1 : A4 = A2
WEND
RETURN
' This file is being presented as an aid to prospective
' AmigaBASIC programmers. I am interested in spreading the
' use of the language because of its extreme versatility.
' I hope that the information presented here has aided you in
' some way.
' If you have any comments or corrections, I can be contacted
' through People Link as AmSoft I.
' If you like to call Amiga BBS's, I can also be reached at:
'     Wonderland BBS (617)-665-3796
'     ZeitGeist BBS (516)-679-3105
```

TAKE THE Q-LINK CHALLENGE!

IF YOU WIN, GET THREE PRIZES!

• A MODEM

• A FOUR-MONTH Q-LINK MEMBERSHIP

• LUCASFILM'S *HABITAT* SOFTWARE

Complete the Challenge of the Month correctly and you will be eligible to win a great prize package!

Each month, 20 individuals will be chosen from the pool of correct Challenge entries and the winners will receive a prize package that includes all this!

- **A free Commodore 300-baud modem (Model 1600).** With this modem, you'll be able to hook up your Commodore 64 or 128 to a telephone and access QuantumLink, the official on-line service for Commodore owners.
- **A free QuantumLink four-month membership.** QuantumLink membership normally costs just $9.95 per month, and includes unlimited use of a base tier of services and one free hour of access each month to special Plus services. As a Challenge winner, you'll get QuantumLink software and your first four months membership free!
- **A free copy of Lucasfilm's *Habitat* software.** *Habitat* is the new multi-player game that utilizes the QuantumLink network to interconnect thousands of Commodore owners from across the country. Participants can quest for hidden treasure, investigate intriguing mysteries, and participate in the ongoing drama of this innovative and exciting graphic adventure.

Consolation Prize!

If the Challenge entry you submit is correct, but your name is not chosen from the pool of correct entries, you will be sent a Q-Link software kit for FREE!

HERE'S HOW TO ENTER!

Simply complete the Challenge of the Month and send it with this coupon to Quantum Computer Services, 8620 Westwood Center Drive, Vienna, VA 22180. Attn: Challenge of the Month #_______ (fill in the correct Challenge of the Month number).

NAME ______________________________

ADDRESS ______________________________

CITY __________ STATE ______ ZIP CODE ______

PHONE NUMBER (_____) ____________________

YOU MUST ANSWER BOTH QUESTIONS TO WIN:

Are you currently a member of QuantumLink?

☐ Yes—if I win, extend my membership for four months?

☐ No—if I win, send me the Q-Link software and free four-month membership. If I get the Challenge correct but don't win, send me a free Q-Link software kit as a consolation prize.

Do you currently own a modem?

☐ Yes—if I win, send me a substitute prize.

☐ No—if I win, send me the modem.

CHALLENGE OF THE MONTH #5

To solve the puzzle, rearrange the letters to form words. The words will form a sentence describing Q-Link.

LNTUAMKQUNI GRNSBI
OMDEOCMRO PECSMOTUR
OT FIEL

A Beginner's Guide to Burst Mode

Continued from pg. 75

Burst Transfer Protocol

The burst transfer protocol is really quite simple. During normal data transfers (those using PRINT#'s, GET#'s, LOAD, SAVE) between a 128 and a 1571 or 1541 drive, a significant fraction of time is taken up by the convoluted path through the KERNAL ROM that must be followed for each byte to be sent and by the signal system used between the sending and receiving devices indicating that a byte has been sent/received and it is time to send/receive the next.

Burst mode eliminates much of this inefficiency by sending data based on a much simpler, and therefore much faster, hardware handshake. Data is exchanged directly between the 1571 and the data register of complex interface adaptor #1 (CIA#1) in the 128 based on a simple signal from the normal serial bus controller (CIA#2) also in the 128.

There are four simple steps to performing a burst mode read operation (data from 1571 to 128): send the appropriate command string to access a burst mode read; initialize the CIA's; read the data; and restore the default I/O devices.

The first step can be performed by either machine language or BASIC statements. As outlined previously, a typical BASIC statement may be open15,8,15,"u0" + chr$(10). This particular example will ask the 1571 to use the QUERY DISK FORMAT command to analyze the format of side 0, track 0 of a disk. The second step is to initialize the interrupt register of CIA#1 and tell the serial bus that you are ready to receive data. It is easiest to do this with machine language (ML). A short ML program can easily be poked into RAM from BASIC or entered with BASIC 7.0's MONITOR command and called with a SYS statement. The ML instructions are

```
   sei
   bit $dc0d
   lda $dd00
   eor #$10
   sta $dd00.
```

If this procedure is being called as a subroutine from either BASIC or ML, then you will need to add an rts after the last instruction to return to the calling program. The first instruction disables the normal processor interupts such as keyboard scanning. This increases the amount of time that the hardware can dedicate to data transfer and eliminates the trapping of keyboard presses which may cause errors during data transfers. The bit instruction is used to reset the interupt control register (ICR) of CIA#1. The final three instructions toggle the state of the acknowledge and ready for data (ARFD) line which is used as a clock during the burst transfer. This is a signal to the 1571 that we are ready to receive data.

The third step is to read the burst data. The ML for reading burst data bytes is quite simple.

```
   lda #$08
wait bit $dc0d
   beq wait
   lda $dd00
   eor #$10
   sta $dd00
   lda $dc0c
```

The first three instructions create a wait loop until bit 3 of CIA#1's ICR is turned on. This indicates that a byte is ready to be received. The label "wait" must be replaced with an absolute address whose value corresponds to where the ML code is located in RAM. The next three instructions toggle the state of the ARFD line causing a data byte to be transferred. The final instruction reads the data byte from the CIA#1 data register. To store this byte, an indexed sta instruction similar to:

```
   ldx #$3f
   stx $ff00
   sta ($fa),y
   ldx #$00
   stx $ff00
```

is normally used (assuming zero page locations $fa and $fb contain the low and high bytes of the data buffer and the y register is used as an index).

In order to use the I/0 and KERNAL routines, the 128 must be set for BANK 15. Unfortunately, this also limits the maximum size of a data buffer to 8K bytes (BANK 0 RAM below $4000 is visiible in BANK 15 also.) This can be overcome by playing with the memory management unit (MMU) configuration register ($ff00—all BANKs) to switch between BANK 15 and BANK 0 on the fly, store the data, then switch back to BANK 15 again. Your machine code must be in an area visible to both BANKs (below $4000 such as the cassette buffer) for this to work. The read routine is often contained in an indexed loop, especially when reading blocks of data. It is important that you keep track of the number of bytes transferred and that your indexing method can handle the number of bytes involved. (Remember that the number of bytes transferred per sector read is 1 plus the number of bytes per sector. MFM format disks may have sector sizes of 128, 256, 512 or 1,024 bytes per sector. FAST LOAD GCR sectors have 254 bytes per sector. The number of bytes transferred for other burst mode commands depends on the command.

The final step after all data have been transferred, processed and stored is to close the disk channel and restore the default I/O. In ML this is done with cli and jsr $ffcc. It is very important to include the cli instruction. This re-enables the processor interupts which were turned off by the initial sei instruction in the second step. The jsr instruction calls the KERNAL CLRCHN routine. That, in a nutshell, is how to read data in burst mode.

Writing in burst mode is only slightly more complicated. There are six basic steps to follow for a burst mode write operation. These are to send the appropriate command string to access a burst mode write; set the serial port to fast output mode; send the data; set the serial port to fast input mode; read the burst status byte (repeat steps 2 to 5 for a multi-sector write); and restore default I/0.

Like other burst mode commands, the WRITE SECTOR command string can be sent via either a BASIC PRINT# statement or a ML JSR CHROUT routine.

The second step is to change the fast serial port direction from the default input mode (data flow from the 1571 to the 128) to output mode (data flow from the 128 to the 1571) and set up the initial clock state. This is best done with a short ML routine using the new 128 KERNAL SPIN/SPOUT routine (serial port input/serial port output) at $ff47. To set the mode to output (SPOUT), the routine is called with the carry flag set

```
   sei
   sec
   jsr $ff47
   lda #$40
   sta clock
```

The last two instructions start the test for the system clock state on a high value. The label clock refers to any usable RAM location (such as zero page $fa to $ff) which is used in subsequent steps as a temporary storage location for testing the state of the system clock.

Once the system has been initialized, the data can be sent. Similar to the read protocol discussed above, data is sent to the 1571 based on a simple toggle handshake using the Acknowledge and Ready for Data (AFRD) line. The procedure is as follows:

```
wait1 lda $dd00
   cmp $dd00
   bne wait1
   eor clock
   and #$40
   beq wait1
   lda ($fa),y
   sta $dc0c
   lda clock
   eor #$40
   sta clock
wait2 lda #$08
   bit $dc0d
   beq wait2
```

The first six instructions form a wait loop until the serial port clock pulse is in the correct phase. The next two instructions retreive the data byte from memory and send it to the serial port. It is assumed that the data buffer address is stored in zero page locations $fa and $fb in standard low-byte, high-byte format.

The next group of three instructions toggles the state of the clock comparison register. The three instructions beginning with the "wait 2" label form a loop until the interrupt control register (ICR) of CIA#1 signals that the transmission of the data byte is complete.

The 1571 returns a status byte after each sector has been written. To read this byte, the fast serial port must be set to the read (SPIN) direction followed by a ready signal to the 1571. This is done with:

```
clc
jsr $ff47
bit $dc0d
lda $dd00
ora #$10
sta $dd00
```

The status byte can then be read with a standard burst mode read as outlined above.

If more sectors are to be written, the whole process starts over again from step 2 (set serial port to SPOUT) until the specified number of sectors has been written. Once all sectors have been written, the final step is to restore default input/output channels in the same manner as described for burst mode reads above.

That is all there is to writing in burst mode. The ML routines for both reading and writing in burst mode can be entered directly on the 128 with its built-in MONITOR command by replacing the labels with absolute addresses. The most convenient location for the ML is the cassette and RS-232 buffers beginning at $0b00 (dec 2816). The combined buffer space gives you 768 bytes for ML code, which is more than adequate for even the most complicated burst mode routines.

Burst Mode and BASIC

If you are willing to forego the extra speed, MFM disks and other burst mode commands can be accessed through BASIC and/or standard ML KERNAL input/output routines. This is done by reading or writing directly from or to the memory of the 1571 drive. A number of important 1571 memory locations are summarized in Table 2.

For example, to read the burst mode status byte, the following statement can be used in a short BASIC program (remember INPUT# and GET# cannot be used in direct mode):

```
OPEN 15,8,15,"M-R"
+CHR$(94)+CHR$(0)+CHR$(1):GET#15,A$
```

The ASCII value of A$ represents the burst mode status byte. Bit 7 of the first byte of the burst mode command string (after the u0) controls the transfer of data between the computer and the 1571. If this bit is set, then data transfer occurs via direct memory read and writes to the 1571 RAM buffers. One sector worth of data for the burst mode READ/WRITE SECTOR commands can be transferred using conventional BASIC or ML through the disk drive command channel. When in burst mode, buffers 0 through 3 (address $300 to 6FF) in the 1571 are used for data storage. Four buffers are allocated for transfer because MFM sectors can be up to 1,024 bytes long. The buffer is filled starting at address $300.

To read an MFM sector in BASIC, the following command sequence can be used.

1. Open the command channel: OPEN 15,8,15
2. Read sector into 1571 buffer: PRINT#15,"U0"+ CHR$(A)+CHR$(track#) +CHR$(sector#)+CHR$(1) where A has a value of 192 to read a sector on side 0 or 208 to read a sector on side 1 of an MFM disk.
3. Extract data from 1571 buffer with a memory-read PRINT#15,"M-R"+ CHR$(BL)+CHR$(BH)+CHR$(0) where BL is the low byte of the 1571 buffer address (0) and BH is the high byte (starts at 3) of the address. The data can then be read with a GET#15,A$ in an indexed loop and stored in 128 memory with a command similar to POKE nnn,ASC(A$). Because only 256 bytes can be read this way at one time, the M-R must be repeated at 1571 buffer addresses of $400, $500 and $600, if the MFM sector size is 512 or 1,024 bytes.
4. CLOSE the command channel: CLOSE 15

The reverse procedure is used to write data on an MFM disk.

1. OPEN the command channel: OPEN 15,8,15
2. Transfer data from the 128 to the 1571 with a memory-write: PRINT#15,"M-W"+CHR$(BL)+CHR$(BH)+CHR$(32); Then PRINT#15 the data in an indexed loop. Because the M-W command is limited to a maximum of only 34 bytes at a time (restricted by the length of the 1571 command input buffer), the M-W must be repeated, incrementing the buffer pointers, BL and BH, each time (as shown above it is easiest to work in increments of 32 rather than 34 because 256= 8 x 32) until one sector has been transferred.
3. Write sector from buffer to disk: PRINT#15,"U0"+CHR$(B)+CHR$(track#) +CHR$(sector#)+CHR$(1) where B has a value of 194 to write to side 0 or 210 to write to side 1 of an MFM disk.

4. CLOSE command channel: CLOSE 15

Although the above examples have been given in BASIC for clarity, the same procedure can be followed using equivalent ML KERNAL calls. By using simple memory read and memory write commands to access the RAM of the 1571 disk drive, even a novice programmer can unlock the secrets of MFM disks with ease! The techniques outlined above can be used with a 64, Plus/4, even a VIC 20, to access the burst mode commands on the 1571.

Program

Listing 1 is a short BASIC program (with ML loader) based on the burst mode FAST LOAD command, which will allow you to compare the speed for reading any file you specify in 1571 fast mode, 1571 burst mode and 1571 slow mode (1541 speed). Try a variety of files of different lengths (both SEQ and PRG). You will see that the longer the file, the greater the speed advantage of burst mode.

In order to use the 1541 slow-mode portion, your test disk should be single-sided. Using a double-sided disk in 1541 mode may create an illegal track and sector error if part of the file is on the second side. The ML portion is poked into the cassette buffer beginning at RAM address 2816 (hex $0b00).

For a look at how the various steps outlined above fit together, dissassemble the ML using the built-in MONITOR command on the 128. The burst fast load routine is in $0b00 to $0b80. Location $0bc6 to 0c01 contains a subroutine to reset and read the CIA real-time clock. This is only used for measuring the transfer speed. This clock is more accurate than BASIC's TI$ (which, by the way, is also turned off during burst mode so can't be used anyway). The jsr's to location $0bc6 and $0be5 may be eliminated if you do not wish to time the transfer. Location $0b90 to $0bc5 contains a conventional read subroutine using the KERNAL. ■

Listing 1

Before typing these programs, read "How to Enter Programs," and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

```
10 PRINT"[CLEAR]1571 FILE READING
   SPEED TEST"'BASF
20 PRINT"[DOWN]BY M. GARAMSZEGHY"
   :PRINT"[DOWN3]PLEASE WAIT"'CBLH
30 FOR I=2816 TO 3072:READ X:POKE I,X
   :NEXT'GQWG
40 INPUT"[DOWN3]FILE TO READ";F$:F$="0
   :"+F$:IF RIGHT$(F$,1)<>"*"THEN
   F$=F$+"*"'KTVO
50 INPUT"[DOWN]FILE TYPE [P]RG OR [S]
   EQ";FT$:IF FT$<>"S"AND
   FT$<>"P"THEN 50'INIQ
60 PRINT"[CLEAR]READING FILE >> ";F$
   :IF FT$="P"THEN C$=CHR$(31)
   :ELSE C$=CHR$(159)'JVRR
70 PRINT"[DOWN2]1571 FAST READ
   : ";"BYTES READ:[SPACE2]";'BCEL
80 OPEN 15,8,15,"U0>M1"
   :REM  SET TO 1571 (FAST) MODE'CDEN
90 OPEN 8,8,8,F$+","+FT$+",R"
   :SYS 2960 ,0,52:GOSUB 170
   :CLOSE 8'HCQP
100 PRINT"[DOWN2]1571 BURST READ
    : ";"BYTES READ:[SPACE2]";'BCDD
110 PRINT#15,"U0"+C$+F$
    :REM SET TO BURST FAST LOAD'EBMG
120 SYS 2816,0,52:GOSUB 170'CNNA
130 PRINT"[DOWN2]1541 SLOW READ
    : ";"BYTES READ:[SPACE2]";'BCYG
140 PRINT#15,"U0>M0":REM  SET TO 1541
    (SLOW) MODE'CXBH
150 OPEN 8,8,8,F$+","+FT$+",R"
    :SYS 2960 ,0,52:GOSUB 170'GAXJ
160 PRINT#15,"U0>M1":DCLOSE:GOTO 40
    :REM  RESET TO 1571 (FAST)
    MODE'EFMM
170 BR=PEEK(250)+PEEK(251)*256-13312
    :PRINT BR'HXDL
180 T=PEEK(3328):S=PEEK(3329)
    :M=PEEK(3330):REM READ CIA CLOCK
    VALUES'HRPS
190 T=INT(T/16)*10+T-INT(T/16)*16
    :REM CONVERT 1/10 SEC BCD TO
    DECIMAL'KSOV
200 S=INT(S/16)*10+S-INT(S/16)*16
    :REM CONVERT SEC BCD TO
    DECIMAL'KOXM
210 M=INT(M/16)*10+M-INT(M/16)*16
    :REM CONVERT MIN BCD TO
    DECIMAL'KOIN
220 SE=T/10+S+M*60:REM ELAPSED TIME
    IN SECONDS FROM CIA CLOCK'GRLN
230 PRINT"READ TIME:[SPACE2]";SE;"SEC,
    [SPACE4]SPEED:[SPACE2]
    "INT(BR/SE)"BYTES/SEC":RETURN'ELGN
240 DATA 133,250,134,251,120,44,12,
    220,32,66,11,32,198,11,234,32'BFBL
250 DATA 59,11,133,252,201,2,176,60,
    160,0,32,59,11,162,63,142'BCQL
260 DATA 0,255,145,250,162,0,142,0,
    255,200,192,254,208,236,152,
    24'BGJN
270 DATA 101,250,133,250,144,2,230,
    251,76,15,11,169,8,44,13,220'BEKN
280 DATA 240,251,173,0,221,73,16,141,
    0,221,173,12,220,96,88,32'BDQO
290 DATA 229,11,96,0,160,0,32,59,11,
    133,252,32,59,11,162,63'BAYP
300 DATA 142,0,255,145,250,162,0,142,
    0,255,200,196,252,208,236,136'BHMI
310 DATA 152,24,101,250,133,250,144,2,
    230,251,76,78,11,0,0,173'BDGI
320 DATA 10,221,141,2,13,173,11,221,
    141,3,13,32,204,255,96,0'BBUJ
330 DATA 133,250,134,251,162,8,32,198,
    11,32,198,255,160,0,32,207'BFXL
340 DATA 255,162,63,142,0,255,145,250,
    162,0,142,0,255,32,183,255'BFSM
350 DATA 208,10,200,192,0,208,231,230,
    251,76,158,11,132,250,32,204'BHWN
360 DATA 255,32,229,11,96,0,173,15,
    221,41,127,141,15,221,173,14'BETN
370 DATA 221,41,128,141,14,221,169,0,
    141,9,221,141,10,221,141,11'BFSO
380 DATA 221,141,8,221,96,173,8,221,
    141,0,13,173,9,221,141,1'BBNP
390 DATA 13,173,10,221,141,2,13,173,
    11,221,141,3,13,32,204,255,96'BGPR
```

END

JIFFIES/THE CLOCK

Continued from pg. 32

20-35	Set and read clock.
50	Locations of hours, minutes, seconds and tenths of seconds.
60	Start address. Change this number to relocate program.
70	Screen memory and color memory.
180	SYS(SA + 104) starts clock running on the screen.
200-240	Set the time.
500-550	Handles input for menu selection.
600-650	Routine used to adjust clock.
700-720	Print program heading.
750-760	Print SYS location to restart clock—end program.
800-810	Move clock around the screen and color memory.
812-850	Print a menu.
900-934	Poke machine language program into memory (will run as it is if SA = 49152).
950-980	Relocate machine-language program depending on SA.
605,901	Turn interrupts off.
985	Turn interrupts back on.

Before typing this program, read "How to Enter Programs" and "How to Use the Magazine Entry Program." The BASIC programs in this magazine are available on disk from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007, 1-800-831-2694.

The Clock

```
10 DEF FN H(DE)=INT(DE/256)'FMGC
15 DEF FN L(DN)=DN-FN H(DN)*256'GPNJ
20 DEF FN S(S)=16*INT(S/10)+S-INT
   (S/10)*10'LTVJ
25 DEF FN R(R)=(((PEEK(R)AND
   112)/1.6)+(PEEK(R)AND 15))AND
   127'KEYP
30 DEF FN PM(S)=PEEK(56331)OR 128'FPQF
35 DEF FN RP(R)=PEEK(56331)AND
   128'FPTK
50 HRS=56331:MIN=56330:SEC=56329
   :TEN=56328'EKDM
60 SA=49152:REM CHANGE SA TO RELOCATE
   ML'CDQK
70 SM=1050:CM=54272:GOSUB 800'DSVI
180 GOSUB 700:GOSUB 900
    :SYS(SA+104)'EPCI
185 GOSUB 800:GOTO 500'CHYK
200 H=FN R(HRS):INPUT"ENTER THE HOUR";
    H'DKOE
202 POKE HRS,FN S(H)+FN RP(0)'ENDD
209 W$="AM":IF FN RP(0) THEN
    W$="PM"'FKNL
210 INPUT"AM OR PM";W$:IF W$<>"AM"
    AND W$<>"PM" THEN 210'ILGH
212 IF W$="PM" THEN POKE HRS,
    FN PM(0)'FLMF
220 M=FN R(MIN):INPUT"ENTER THE
    MINUTE";M'DKPG
222 POKE MIN,FN S(M)'CIKD
230 S=FN R(SEC):INPUT"ENTER THE
    SECOND";S'DKVH
232 POKE SEC,FN S(S)'CIHE
240 POKE TEN,0'BFRB
490 GOSUB 812'BDOI
500 GET K$'BCQA
510 IF K$=CHR$(136)THEN 750'EKKE
512 IF K$=CHR$(19) THEN C=1050'FLSH
515 IF K$=CHR$(29) THEN C=SM+1'GKTL
520 IF K$=CHR$(157)THEN C=SM-1'GLUH
525 IF K$=CHR$(17) THEN C=SM+40'GLRM
530 IF K$=CHR$(145)THEN C=SM-40'GMSI
532 IF K$=CHR$(133)THEN GOSUB 700
    :GOTO 200'GOWK
533 IF K$=CHR$(135)THEN GOSUB 600'FKNK
535 IF K$=CHR$(134)THEN POKE SA+92,
    (PEEK(SA+92)+1)AND 15'KXHS
540 IF C>1023 AND C<2011 THEN SM=C
    :GOSUB 800:C=0'IURM
550 GOTO 500'BDEF
600 B=B+1 AND 3:IF B=0 THEN 70'GJIF
605 POKE U,PEEK(U)AND 254'DIBI
610 B(1)=29:B(2)=38:B(3)=44:L(1)=10
    :L(2)=7:L(3)=4'GOUN
615 POKE SA+29,240'CICJ
620 POKE SA+30,B(B)'CJGF
625 POKE SA+94,L(B)'CJBK
650 CM=54271:GOSUB 800:GOTO 985'DPNK
700 PRINT CHR$(147):PRINT
    :PRINT TAB(16)"THE CLOCK"'FKEI
710 PRINT TAB(10)"BY BENNETT COOKSON
    JR.":PRINT:PRINT'EFLL
720 RETURN'BAQD
750 PRINT:PRINT"SYS";SA+104;
    "TO RESTART AFTER STOP/RESTORE"'D
    IET
760 END'BACH
800 RESTORE:DATA 8,17,23,45,51,57,
    89'CUCH
805 FOR X=1 TO 7:READ I
    :POKE SA+I,FN L(SM)
    :POKE SA+I+1,FN H(SM):NEXT'MCAV
810 POKE SA+96,FN L(SM+CM)
    :POKE SA+97,FN H(SM+CM)'IADO
812 GOSUB 700'BDKG
815 PRINT"F1 TO SET TIME":PRINT'CBMN
820 PRINT"CURSOR CONTROLS TO MOVE
    CLOCK":PRINT'CBGN
822 PRINT"HOME TO MOVE CLOCK TO TOP"
    :PRINT'CBLO
825 PRINT"F3 TO CHANGE COLOR"
    :PRINT'CBCP
830 PRINT"F5 TO ADJUST TYPE"
    :PRINT'CBFL
840 PRINT"F7 TO END PROGRAM"
    :PRINT'CBHL
850 RETURN'BAQH
900 I=SA'BDRE
901 U=56334:POKE U,PEEK(U)AND 254
    :REM TURN INTERRUPTS OFF'FIDQ
902 READ A:IF A=256 THEN 950'EJWJ
904 POKE I,A:I=I+1:GOTO 902'ELEM
906 DATA 170,74,74,74,74,9,48,153'BYGO
908 DATA 26,4,200,138,41,15,9,48'BXPP
910 DATA 153,26,4,200,169,58,153,
    26'BBOJ
912 DATA 4,200,96,160,0,32,20,192'BYWL
```

Continued on pg. 124

HOW TO ENTER PROGRAMS

The programs which appear in this magazine have been run, tested and checked for bugs and errors. After a program is tested, it is printed on a letter quality printer with some formatting changes. This listing is then photographed directly and printed in the magazine. Using this method ensures the most error-free program listings possible.

Whenever you see a word inside brackets, such as [DOWN], the word represents a keystroke or series of keystrokes on the keyboard. The word [DOWN] would be entered by pressing the cursor-down key. If multiple keystrokes are required, the number will directly follow the word. For example, [DOWN4] would mean to press the cursor-down key four times. If there are multiple words within one set of brackets, enter the keystrokes directly after one another. For example, [DOWN,RIGHT2] would mean to press the cursor-down key once and then the cursor-right key twice. Note: Do not enter the commas.

In addition to these graphic symbols, the keyboard graphics are all represented by a word and a letter. The word is either SHFT or CMD and represents the SHIFT key or the Commodore key. The letter is one of the letters on the keyboard. The combination [SHIFT E] would be entered by holding down the SHIFT key and pressing the E. A number following the letter tells you how many times to type the letter. For example, [SHFT A4,CMD B3] would mean to hold the SHIFT key and press the A four times, then hold down the Commodore key and press the B three times.

The following chart tells you the keys to press for any word or words inside of brackets. Refer to this chart whenever you aren't sure what keys to press. The little graphic next to the keystrokes shows you what you will see on the screen.

Syntax Error

This is by far the most common error encountered while entering a program. Usually (sorry folks) this means that you have typed something incorrectly on the line the syntax error refers to. If you get the message "?Syntax Error Break In Line 270," type LIST 270 and press RETURN. This will list line 270 to the screen. Look for any non-obvious mistakes like a zero in place of an O or vice-versa. Check for semicolons and colons reversed and extra or missing parentheses. All of these things will cause a syntax error.

There is only one time a syntax error will tell you the wrong line to look at. If the line the syntax error refers to has a function call (e.g., FN A(3)), the syntax error may be in the line that defines the function, rather than the line named in the error message. Look for a line near the beginning of the program (usually) that has DEF FN A(X) in it with an equation following it. Look for a typo in the equation part of this definition.

Illegal Quantity Error

This is another common error message. This can also be caused by a typing error, but it is a little harder to find. Once again, list the line number that the error message refers to. There is probably a poke statement on this line. If there is, then the error is referring to what is trying to be poked. A number must be in the range of zero to 255 to be poke-able. For example, the statement POKE 1024,260 would produce an illegal quantity error because 260 is greater than 255.

Most often, the value being poked is a variable (A,X...). This error is telling you that this variable is out of range. If the variable is being read from data statements, then the problem is somewhere in the data statements. Check the data statements for missing commas or other typos.

If the variable is not coming from data statements, then the problem will be a little harder to find. Check each line that contains the variable for typing mistakes.

Out Of Data Error

This error message is always related to the data statements in a program. If this error occurs, it means that the program has run out of data items before it was supposed to. It is usually caused by a problem or typo in the data statements. Check first to see if you have left out a whole line of data. Next, check for missing commas between numbers. Reading data from a page of a magazine can be a strain on the brain, so use a ruler or a piece of paper or anything else to help you keep track of where you are as you enter the data.

Other Problems

It is important to remember that the 64 and the PET/CBM computers will only accept a line up to 80 characters long. The VIC 20 will accept a line up to 88 characters long and the 128 a line up to 160 characters long. Sometimes you will find a line in a program that runs over this number of characters. This is not a mistake in the listing. Sometimes programmers get so carried away crunching programs that they use abbreviated commands to get more than the standard number of characters on one line.

"[HOME]" = UNSHIFTED CLR/ HOME
"[CLEAR]" = SHIFTED CLR/HOME
"[DOWN]" = CURSOR DOWN
"[UP]" = CURSOR UP
"[RIGHT]" = CURSOR RIGHT
"[LEFT]" = CURSOR LEFT
"[RVS]" = CONTROL 9
"[RVOFF]" = CONTROL 0
"[BLACK]" = CONTROL 1
"[WHITE]" = CONTROL 2
"[RED]" = CONTROL 3
"[CYAN]" = CONTROL 4

"[PURPLE]" = CONTROL 5
"[GREEN]" = CONTROL 6
"[BLUE]" = CONTROL 7
"[YELLOW]" = CONTROL 8
"[ORANGE]" = COMMODORE 1
"[BROWN]" = COMMODORE 2
"[L. RED]" = COMMODORE 3
"[GRAY1]" = COMMODORE 4
"[GRAY2]" = COMMODORE 5
"[L. GREEN]" = COMMODORE 6
"[L. BLUE]" = COMMODORE 7
"[GRAY3]" = COMMODORE 8

"[F1]" = F1
"[F2]" = F2
"[F3]" = F3
"[F4]" = F4
"[F5]" = F5
"[F6]" = F6
"[F7]" = F7
"[F8]" = F8
"[POUND]" = ENGLISH POUND
"[SHFT ^]" = PI SYMBOL
"[^]" = UP ARROW

GRAPHIC SYMBOLS WILL BE REPRESENTED AS EITHER THE LETTERS SHFT (SHIFT) AND A KEY ("[SHFT Q,SHFT J,SHFT D,SHFT S]") OR THE LETTERS CMDR (COMMODORE) AND A KEY ("[CMDR Q,CMDR G,COMDR Y,CMDR H]"). IF A SYMBOL IS REPEATED, THE NUMBER OF REPITITIONS WILL BE DIRECTLY AFTER THE KEY AND BEFORE THE COMMA ("[SPACE3,SHFT S4,CMDR M2]").

HOW TO ENTER PROGRAMS

You can enter these lines by abbreviating the commands when you enter the line. The abbreviations for BASIC commands are in your user guide.

If you type a line that is longer than the acceptable number of characters, the computer will act as if everything is ok, until you press RETURN. Then, a syntax error will be displayed (without a line number). Many people write that the computer gives them a syntax error when they type the line, or that the computer refuses to accept a line. Both of these problems are results of typing a line that has too many characters.

The Program Won't Run!!

This is the hardest of problems to resolve; no error message is displayed, but the program just doesn't run. This can be caused by many small mistakes typing a program in. First check that the program was written for the computer you are using. Check to see if you have left out any lines of the program. Check each line of the program for typos or missing parts. Finally, press the RUN/STOP key while the program is "running." Write down the line the program broke at and try to follow the program backwards from this point, looking for problems.

If All Else Fails

You've come to the end of your rope. You can't get the program to run and you can't find any errors in your typing. What do you do? As always, we suggest that you try a local user group for help. In a group of even just a dozen members, someone is bound to have typed in the same program. The user group may also have the program on a library disk and be willing to make a copy for you. For $9.95 per issue, you can also get all the BASIC programs in each issue, as well, from Loadstar, P.O. Box 30007, Shreveport, LA 71130-0007.

If you do get a working copy, be sure to compare it to your own version so that you can learn from your errors and increase your understanding of programming.

If you live in the country, don't have a local user group, or you simply can't get any help, write to us. If you do write to us, include the following information about the program you are having problems with:

- The name of the program
- The issue of the magazine it was in
- The computer you are using
- Any error messages and the line numbers
- Anything displayed on the screen
- A printout of your listing (if possible)

All of this information is helpful in answering your questions about why a program doesn't work. A letter that simply states "I get an error in line 250 whenever I run the program" doesn't give us much to go on. Send your questions to:

Commodore Magazines
1200 Wilson Drive
West Chester, PA 19380
ATTN: Program Problem

HOW TO USE THE MAGAZINE ENTRY PROGRAMS

The Magazine Entry Programs on the next pages are two BASIC machine language programs that will assist you in entering the programs in this magazine correctly. There are versions for both the Commodore 64 and the Commodore 128. Once the program is in place, it works its magic without you having to do anything else. The program will not let you enter a line if there is a typing mistake on it, and better yet, it identifies the kind of error for you.

Getting Started

Type in the Magazine Entry Program carefully and save it as you go along (just in case). Once the whole program is typed in, save it again on tape or disk. Now RUN the program. The word POKING will appear on the top of the screen with a number. The number will increment from 49152 up to 49900 (4864-5545 on the 128) and just lets you know that the program is running. If everything is ok, the program will finish running and say DONE. Then type NEW. If there is a problem with the data statements, the program will tell you where to find the problem. Otherwise the program will say "mistake in data statements." Check to see if commas are missing, or if you have used periods instead of commas. Also check the individual data items.

Once the program has run, it is in memory ready to go. To activate the program type SYS49152 (SYS4864 on the 128), and press RETURN. You are now ready to enter the programs from the magazine. To disable the Entry Program, just type KILL (RETURN) on the 64 or SYS4867 on the 128.

The checksums for each line are the same for both the 64 and 128, so you can enter your 64 programs on the 128 if you'd like.

Typing the Programs

All the BASIC program listings in this magazine that are for the 64 or 128 have an apostrophe followed by four letters at the end of the line (e.g., 'ACDF). If you plan to use the Magazine Entry Program to enter your programs, the apostrophe and letters **should** be entered along with the rest of the line. This is a checksum that the Magazine Entry Program uses.

Enter the line and the letters at the end and then press RETURN, just as you normally would.

If the line is entered correctly, a bell is sounded and the line is entered into the computer's memory (without the characters at the end).

If a mistake was made while entering the line, a noise is sounded and an error message is displayed. Read the error message, then press any key to erase the message and correct the line.

IMPORTANT

If the Magazine Entry Program sees a mistake on a line, it **does not** enter that line into memory. This makes it impossible to enter a line incorrectly.

Error Messages and What They Mean

There are five error messages that the Magazine Entry Program uses. Here they are, along with what they mean and how to fix them.

Continued next page

HOW TO USE THE MAGAZINE ENTRY PROGRAMS

NO CHECKSUM: This means that you forgot to enter the apostrophe and the four letters at the end of the line. Move the cursor to the end of the line you just typed and enter the checksum.

QUOTE: This means that you forgot (or added) a quote mark somewhere in the line. Check the line in the magazine and correct the quote.

KEYWORD: This means that you have either forgotten a command or spelled one of the BASIC keywords (GOTO, PRINT. .) incorrectly. Check the line in the magazine again and check your spelling.

OF CHARACTERS: This means that you have either entered extra characters or missed some characters. Check the line in the magazine again. This error message will also occur if you misspell a BASIC command, but create another keyword in doing so. For example, if you misspell PRINT as PRONT, the 64 sees the letter P and R, the BASIC keyword ON and then the letter T. Because it sees the keyword ON, it thinks you've got too many characters, instead of a simple misspelling. Check spelling of BASIC commands if you can't find anything else wrong.

UNIDENTIFIED: This means that you have either made a simple spelling error, you typed the wrong line number, or you typed the checksum incorrectly. Spelling errors could be the wrong number of spaces inside quotes, a variable spelled wrong, or a word misspelled. Check the line in the magazine again and correct the mistake.

MAGAZINE ENTRY PROGRAM—64

The Magazine Entry Programs are available on disk, along with the other programs in this magazine, for $9.95. To order, contact Loadstar at 1-800-831-2694.

```
10 PRINT"[CLEAR]POKING -";
20 P=49152 :REM $C000  (END AT
   49900/$C2EC)
30 READ A$:IF A$="END"THEN 110
40 L=ASC(MID$(A$,2,1))
50 H=ASC(MID$(A$,1,1))
60 L=L-48:IF L>9 THEN L=L-7
70 H=H-48:IF H>9 THEN H=H-7
80 PRINT"[HOME,RIGHT12]"P;
90 IF H>15 OR L>15 THEN PRINT
   :PRINT"DATA ERROR IN LINE";
   1000+INT((P-49152)/8):STOP
100 B=H*16+L:POKE P,B:T=T+B:P=P+1
   :GOTO 30
110 IF T<>86200 THEN PRINT
   :PRINT"MISTAKE IN DATA --> CHECK
   DATA STATEMENTS":END
120 PRINT"DONE":END
1000 DATA 4C,1F,C0,00,00,00,00,00
1001 DATA 00,00,00,00,00,0D,00,21
1002 DATA C1,27,C1,2F,C1,3F,C1,4C
1003 DATA C1,EA,EA,EA,4C,54,C0,A2
1004 DATA 05,BD,19,C0,95,73,CA,10
1005 DATA F8,60,60,A0,03,B9,00,02
1006 DATA D9,04,C1,D0,F5,88,10,F5
1007 DATA A0,05,B9,A2,E3,99,73,00
1008 DATA 88,10,F7,A9,00,8D,18,D4
1009 DATA 4C,EF,C0,E6,7A,D0,02,E6
1010 DATA 7B,4C,79,00,A5,9D,F0,F3
1011 DATA A5,7A,C9,FF,D0,ED,A5,7B
1012 DATA C9,01,D0,E7,20,2B,C0,AD
1013 DATA 00,02,20,74,C0,90,DC,A0
1014 DATA 00,4C,A9,C1,C9,30,30,06
1015 DATA C9,3A,10,02,38,60,18,60
1016 DATA C8,B1,7A,C9,20,D0,03,C8
1017 DATA D0,F7,B1,7A,60,18,C8,B1
1018 DATA 7A,F0,37,C9,22,F0,F5,6D
1019 DATA 03,C0,8D,03,C0,AD,04,C0
1020 DATA 69,00,8D,04,C0,4C,8E,C0
1021 DATA 18,6D,05,C0,8D,05,C0,90
1022 DATA 03,EE,06,C0,EE,09,C0,4C
1023 DATA CE,C1,18,6D,08,C0,8D,08
1024 DATA C0,90,03,EE,07,C0,EE,0A
1025 DATA C0,60,0A,A8,B9,0F,C0,85
1026 DATA FB,B9,10,C0,85,FC,A0,00
1027 DATA A9,12,20,D2,FF,B1,FB,F0
1028 DATA 06,20,D2,FF,C8,D0,F6,20
1029 DATA BC,C2,20,E4,FF,F0,FB,A0
1030 DATA 18,B9,08,C1,20,D2,FF,88
1031 DATA 10,F7,68,68,A9,00,8D,00
1032 DATA 02,4C,74,A4,4B,49,4C,4C
1033 DATA 91,91,0D,20,20,20,20,20
1034 DATA 20,20,20,20,20,20,20,20
1035 DATA 20,20,20,20,20,20,20,91
1036 DATA 0D,51,55,4F,54,45,00,4B
1037 DATA 45,59,57,4F,52,44,00,23
1038 DATA 20,4F,46,20,43,48,41,52
1039 DATA 41,43,54,45,52,53,00,55
1040 DATA 4E,49,44,45,4E,54,49,46
1041 DATA 49,45,44,00,4E,4F,20,43
1042 DATA 48,45,43,4B,53,55,4D,00
1043 DATA C8,B1,7A,D0,FB,84,FD,C0
1044 DATA 09,10,03,4C,84,C1,88,88
1045 DATA 88,88,88,B1,7A,C9,27,D0
1046 DATA 13,A9,00,91,7A,C8,A2,00
1047 DATA B1,7A,9D,3C,03,C8,E8,E0
1048 DATA 04,D0,F5,60,A9,04,4C,CA
1049 DATA C0,A0,00,B9,00,02,99,40
1050 DATA 03,F0,F0,C8,D0,F5,A0,00
1051 DATA B9,40,03,F0,E6,99,00,02
1052 DATA C8,D0,F5,20,96,C1,4C,12
1053 DATA C2,A0,09,A9,00,99,03,C0
1054 DATA 8D,3C,03,88,10,F7,A9,80
1055 DATA 85,02,A0,00,20,58,C1,20
1056 DATA 89,C1,20,ED,C1,E6,7A,E6
1057 DATA 7B,20,7C,A5,A0,00,20,80
1058 DATA C0,F0,D0,24,02,F0,06,4C
1059 DATA A8,C0,4C,CE,C1,C9,22,D0
1060 DATA 06,20,8D,C0,4C,CE,C1,20
1061 DATA BA,C0,4C,CE,C1,A0,00,B9
1062 DATA 00,02,20,74,C0,C8,90,0A
1063 DATA 18,6D,07,C0,8D,07,C0,4C
1064 DATA EF,C1,88,A2,00,B9,00,02
1065 DATA 9D,00,02,F0,04,E8,C8,D0
1066 DATA F4,60,18,AD,09,C0,69,41
1067 DATA 8D,09,C0,38,AD,0A,C0,E9
1068 DATA 19,90,06,8D,0A,C0,4C,1C
1069 DATA C2,AD,0A,C0,69,41,8D,0A
```

```
1Ø7Ø DATA CØ,AD,Ø3,CØ,6D,Ø5,CØ,48
1Ø71 DATA AD,Ø4,CØ,6D,Ø6,CØ,8D,ØC
1Ø72 DATA CØ,68,6D,Ø8,CØ,8D,ØB,CØ
1Ø73 DATA AD,ØC,CØ,6D,Ø7,CØ,8D,ØC
1Ø74 DATA CØ,38,E9,19,9Ø,Ø6,8D,ØC
1Ø75 DATA CØ,4C,52,C2,AD,ØC,CØ,69
1Ø76 DATA 41,8D,ØC,CØ,AD,ØB,CØ,E9
1Ø77 DATA 19,9Ø,Ø6,8D,ØB,CØ,4C,67
1Ø78 DATA C2,AD,ØB,CØ,69,41,8D,ØB
1Ø79 DATA CØ,AØ,Ø1,AD,Ø9,CØ,CD,3C
1Ø8Ø DATA Ø3,DØ,2Ø,C8,AD,ØA,CØ,CD
1Ø81 DATA 3D,Ø3,DØ,17,C8,AD,ØB,CØ
1Ø82 DATA CD,3E,Ø3,DØ,ØE,AD,ØC,CØ
1Ø83 DATA CD,3F,Ø3,DØ,Ø6,2Ø,CC,C2
1Ø84 DATA 4C,4B,CØ,98,48,68,4C,CA
1Ø85 DATA CØ,A9,2Ø,8D,ØØ,D4,8D,Ø1
1Ø86 DATA D4,A9,Ø9,8D,Ø5,D4,A9,ØF
1Ø87 DATA 8D,18,D4,6Ø,2Ø,A9,C2,A9
1Ø88 DATA 81,2Ø,DF,C2,A9,8Ø,2Ø,DF
1Ø89 DATA C2,4C,D9,C2,2Ø,A9,C2,A9
1Ø9Ø DATA 11,2Ø,DF,C2,A9,1Ø,2Ø,DF
1Ø91 DATA C2,A9,ØØ,8D,Ø4,D4,6Ø,8D
1Ø92 DATA Ø4,D4,A2,7Ø,AØ,ØØ,88,DØ
1Ø93 DATA FD,CA,DØ,FA,6Ø,END
```

END

MAGAZINE ENTRY PROGRAM—128

```
5 TRAP 2ØØ
1Ø PRINT"[CLEAR]POKING -";
2Ø P=4864 :REM $13ØØ  (END AT
   5545/$15A9)
3Ø READ A$:IF A$="END"THEN 11Ø
8Ø PRINT"[HOME,RIGHT12]"P;
1ØØ B=DEC(A$):POKE P,B:T=T+B:P=P+1
    :GOTO 3Ø
11Ø IF T<>59311 THEN PRINT
    :PRINT"MISTAKE IN DATA --> CHECK
    DATA STATEMENTS":END
12Ø PRINT"DONE":END
2ØØ PRINT:PRINT"DATA ERROR IN LINE";
    1ØØØ+INT((P-4864)/8):END
1ØØØ DATA 4C,1E,13,4C,3A,13,ØØ,ØØ
1ØØ1 DATA 8E,ØØ,F7,ØØ,42,41,51,57
1ØØ2 DATA ØD,ØØ,ØD,43,Ø8,14,ØE,14
1ØØ3 DATA 16,14,26,14,33,14,A9,ØØ
1ØØ4 DATA 8D,ØØ,FF,AD,Ø4,Ø3,8D,12
1ØØ5 DATA 13,AD,Ø5,Ø3,8D,13,13,A2
1ØØ6 DATA 4A,AØ,13,8E,Ø4,Ø3,8C,Ø5
1ØØ7 DATA Ø3,6Ø,AD,12,13,8D,Ø4,Ø3
1ØØ8 DATA AD,13,13,8D,Ø5,Ø3,6Ø,6C
1ØØ9 DATA 12,13,A5,7F,DØ,F9,AD,ØØ
1Ø1Ø DATA Ø2,2Ø,5B,13,9Ø,F1,AØ,ØØ
1Ø11 DATA 4C,6F,14,C9,3Ø,3Ø,Ø6,C9
1Ø12 DATA 3A,1Ø,Ø2,38,6Ø,18,6Ø,C8
1Ø13 DATA B1,3D,C9,2Ø,DØ,Ø3,C8,DØ
1Ø14 DATA F7,B1,3D,6Ø,18,C8,B1,3D
1Ø15 DATA FØ,35,C9,22,FØ,F5,6D,Ø6
1Ø16 DATA 13,8D,Ø6,13,AD,Ø7,13,69
1Ø17 DATA ØØ,8D,Ø7,13,4C,75,13,18
1Ø18 DATA 6D,Ø8,13,8D,Ø8,13,9Ø,Ø3
1Ø19 DATA EE,Ø9,13,EE,ØC,13,6Ø,18
1Ø2Ø DATA 6D,ØB,13,8D,ØB,13,9Ø,Ø3
1Ø21 DATA EE,ØA,13,EE,ØD,13,6Ø,ØA
1Ø22 DATA A8,B9,14,13,85,FB,B9,15
1Ø23 DATA 13,85,FC,AØ,ØØ,8C,ØØ,FF
1Ø24 DATA A9,12,2Ø,D2,FF,B1,FB,FØ
1Ø25 DATA Ø6,2Ø,D2,FF,C8,DØ,F6,2Ø
1Ø26 DATA 79,15,2Ø,A3,15,2Ø,E4,FF
1Ø27 DATA FØ,FB,AØ,1B,B9,EF,13,2Ø
1Ø28 DATA D2,FF,88,1Ø,F7,68,68,A9
1Ø29 DATA ØØ,8D,ØØ,Ø2,4C,B7,4D,91
1Ø3Ø DATA 91,ØD,2Ø,2Ø,2Ø,2Ø,2Ø,2Ø
1Ø31 DATA 2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,2Ø
1Ø32 DATA 2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,91,ØD
1Ø33 DATA 51,55,4F,54,45,ØØ,4B,45
1Ø34 DATA 59,57,4F,52,44,ØØ,23,2Ø
1Ø35 DATA 4F,46,2Ø,43,48,41,52,41
1Ø36 DATA 43,54,45,52,53,ØØ,55,4E
1Ø37 DATA 49,44,45,4E,54,49,46,49
1Ø38 DATA 45,44,ØØ,4E,4F,2Ø,43,48
1Ø39 DATA 45,43,4B,53,55,4D,ØØ,C8
1Ø4Ø DATA B1,3D,DØ,FB,CØ,Ø6,1Ø,Ø3
1Ø41 DATA 4C,69,14,88,88,88,88,88
1Ø42 DATA B1,3D,C9,27,DØ,13,A9,ØØ
1Ø43 DATA 91,3D,C8,A2,ØØ,B1,3D,9D
1Ø44 DATA ØØ,ØB,C8,E8,EØ,Ø4,DØ,F5
1Ø45 DATA 6Ø,4C,5C,15,4C,C5,14,AØ
1Ø46 DATA Ø9,A9,ØØ,99,Ø6,13,8D,ØØ
1Ø47 DATA ØB,88,1Ø,F7,A9,8Ø,85,FD
1Ø48 DATA AØ,ØØ,2Ø,3F,14,2Ø,AE,14
1Ø49 DATA 2Ø,ØD,43,84,FA,AØ,FF,2Ø
1Ø5Ø DATA 67,13,FØ,D8,24,FD,FØ,Ø6
1Ø51 DATA 2Ø,8F,13,4C,8F,14,C9,22
1Ø52 DATA DØ,Ø6,2Ø,74,13,4C,8F,14
1Ø53 DATA 2Ø,9F,13,4C,8F,14,AØ,ØØ
1Ø54 DATA B9,ØØ,Ø2,2Ø,5B,13,C8,9Ø
1Ø55 DATA ØA,18,6D,ØA,13,8D,ØA,13
1Ø56 DATA 4C,BØ,14,88,6Ø,18,AD,ØC
1Ø57 DATA 13,69,41,8D,ØC,13,38,AD
1Ø58 DATA ØD,13,E9,19,9Ø,Ø6,8D,ØD
1Ø59 DATA 13,4C,CF,14,AD,ØD,13,69
1Ø6Ø DATA 41,8D,ØD,13,AD,Ø6,13,6D
1Ø61 DATA Ø8,13,48,AD,Ø7,13,6D,Ø9
1Ø62 DATA 13,8D,ØF,13,68,6D,ØB,13
1Ø63 DATA 8D,ØE,13,AD,ØF,13,6D,ØA
1Ø64 DATA 13,8D,ØF,13,38,E9,19,9Ø
1Ø65 DATA Ø6,8D,ØF,13,4C,Ø5,15,AD
1Ø66 DATA ØF,13,69,41,8D,ØF,13,AD
1Ø67 DATA ØE,13,E9,19,9Ø,Ø6,8D,ØE
1Ø68 DATA 13,4C,1A,15,AD,ØE,13,69
1Ø69 DATA 41,8D,ØE,13,AØ,Ø1,AD,ØC
1Ø7Ø DATA 13,CD,ØØ,ØB,DØ,2Ø,C8,AD
1Ø71 DATA ØD,13,CD,Ø1,ØB,DØ,17,C8
1Ø72 DATA AD,ØE,13,CD,Ø2,ØB,DØ,ØE
1Ø73 DATA AD,ØF,13,CD,Ø3,ØB,DØ,Ø6
1Ø74 DATA 2Ø,89,15,A4,FA,6Ø,98,48
1Ø75 DATA 68,4C,AF,13,A9,Ø4,4C,AF
1Ø76 DATA 13,A9,ØØ,8D,ØØ,FF,A9,2Ø
1Ø77 DATA 8D,ØØ,D4,8D,Ø1,D4,A9,Ø9
1Ø78 DATA 8D,Ø5,D4,A9,ØF,8D,18,D4
1Ø79 DATA 6Ø,2Ø,61,15,A9,81,2Ø,9C
1Ø8Ø DATA 15,A9,8Ø,2Ø,9C,15,4C,96
1Ø81 DATA 15,2Ø,61,15,A9,11,2Ø,9C
1Ø82 DATA 15,A9,1Ø,2Ø,9C,15,A9,ØØ
1Ø83 DATA 8D,Ø4,D4,6Ø,8D,Ø4,D4,A2
1Ø84 DATA 7Ø,AØ,ØØ,88,DØ,FD,CA,DØ
1Ø85 DATA FA,6Ø,END
```

END

USER GROUPS

Although there are almost 1000 known Commodore user groups nationwide and around the world, this list includes only those that have been officially recognized by Commodore as Approved User Groups. If your group would like to apply for Approved status, contact Pete Baczor, User Group Coordinator, at Commodore Business Machines, 1200 Wilson Drive, West Chester, PA 19380.

Commodore user groups provide invaluable assistance to Commodore computerists. If you are looking for people who share your computing interests, or if you need help getting started with your computer, contact the group near you.

ALABAMA

Birmingham
Birmingham Commodore Club
P O Box 59564
Bob Tatum

Decatur
Valley Commodore User Group
915 Way Thru the Woods SW
David W Nelson

Florence
Shoals Commodore Users Group
114 Van Fleet Dr
Russel C Brockwell

Huntsville
Computeen Computer Club
10118 Shades Road
Bryan Jordan

Montgomery
M A C K S
P O Box 210126
Charles Russell

Scottsboro
Scottsboro C-64 Users Group
Route #5, Box 255
Richard Radon

ALASKA

Anchorage
Anchorage Commodore User Group
P O Box 104615
Troy L Crow, Jr

ALBERTA CANADA

Hinton
Hinton Computer Club
P O Box 2431
K Bartlett

ARIZONA

Chandler
Arizona VIC & 64 Users
904 W Marlboro Cir
Thomas A Monson

Chino Valley
Prescott Area Commodore Club
1631 N Canfield Ave
Richard Pearce

Globe
Gila Hackers
Route #1, Box 34
Paul R Machula

Sierra Vista
Thunder Mountain Commodore User Group
P O Box 1796
Michael Magatagan

Tucson
Catalina Commodore Computer Club
P O Box 32548
George Pope

ARKANSAS

North Little Rock
River City Commodore Computer Club
P O Box 4298
G Gordon Yoder, Jr

Pine Bluff
Commodore Computer Club of Pine Bluff
P O Box 1083r
Bruce Ashcraft

BRITISH COLUMBIA CANADA

Victoria
Juan de Fuca C64 Users' Group
442 Tipton Rd
Bryan Wilkinson

CALIFORNIA

Auburn
Auburn Commodore Computer Club
P O Box 4270
Pat Strub

Concord
Commodore Connection
P O Box 272106
Richard Moxley

Concord
Diablo Valley Commodore User Group
P O Box 27155
Richard Jensen

Fullerton
HFEA Commodore Computer Club
P O Box 3310
Ed Charles

Huntington Beach
West Orange County Commodore UG
20311 Ravenwood Lane
Milton Brown

Los Osos
San Luis Obispo Commodore Club
1766 Ninth Street
Joan Rinehart

Newport Beach
Ford Aerospace Commodore Users Group
Ford Rd, Box A, Mail Stop #2/208
Joseph R Pasek

Oceanside
Oceana-64
1004 Plover Way
Sam Brooks

Paso Robles
Simply Users of Computers Combinin
301 Veronica Drive
Wayne Weichel

Sacramento
Sacramento Commodore Users Group
5900 Bamford Dr
Lanny Hartzberg

San Diego
San Diego Commodore Users Group
P O Box 86531
Jane Campbell

San Rafael
Marin Commodore Computer Club
665 Las Colindas Rd
Elmer E Johnson

Santa Barbara
Santa Barbara Commodore Users Group
4520 Via Vistosa
Walter Hauz

Stockton
Stockton Commodore User Group
2929 Calariva Dr
Rita McCann

Studio City
Southern CA 64 Users Group
3852 Riverton
Joe Allain

Ventura
64/20 Club
6464 Sheawater St
Greg Burns

COLORADO

Aurora
Colorado PET Users Group
676 S Quentin St
Jon C Almon

Central City
Black-Cat Software
P O Box 144
Jason M Murphy

Grand Junction
Western Slope Commodore User Group
P O Box 4142
Joe Gardner

Lakewood
Commodore Condor Club
1680 Lewis St
Ronald Musich

CONNECTICUT

Danbury
Fairfield County Commodore User Group
P O Box 212
Kenneth H Hottes

East Hartford
Hartford County Commodore Users Group
P O Box 8553
J Russell Curtiss

North Haven
Greater New Haven Commodore User Group
P O Box 796
Ray Goudneau

Stamford
Stamford Area Commodore Society
P O Box 1337
Richard P Mozzer

Vernon
Capitol Region Commodore Club
P O Box 2372
Peter Brochu

DELAWARE

Newark
Newark Commodore Users Group
210 Durso Dr
Robert E Black, Jr

DISTRICT OF COLUMBIA

Pentagon—Wash
PENTAF Commodore 64 Club
1947th HSG-MWR, AF Rec Svcs

FLORIDA

Beverly Hills
Citrus County Commodore Club
P O Box 503
Walt Rogers

Bradenton
Manasota Commodore User Group
916 E 35th Ave Dr
Robert O Bronson

Englewood
RAM ROM 64
P O Box 1369
Clyde S Davies

Ft Walton Beach
Okaloosa Commodore User's Group
412-A Cobia Avenue
Calvin Jones

Gainesville
Gainesville Commodore Users Group
P O Box 14716
Drew Hurley

Jacksonville
UNF Commodore Computer Club
University of Northern Florida
Jay Huebner

Lakeland
Lakeland Police User Group
2332 Crystal Park North
Joseph M Salvadore

Melbourne
EL-Shift-OH
P O Box 361348
Ted Semanism

Miami
M I C E
11110 Bird Rd
Ben C Demby, Jr

Miami
Miami 2064
12911 SW 49th Street
Jim Juftman

Miami Springs
Miami Springs Hialeah User Group
517 Wren Avenue
Charles P Taffinder

Ocala
Tri-County Commodore Users Group
P O Box 1151
Helen Serine

Orange Park
Commodore Computer Club
3931 Collins Rd
Loring Pitts

Orlando
Central FL Commodore Users Group
P O Box 7326
Thurmon Lawson

Ozona
Suncoast 64's
P O Box 6628
Robert P Deley

Pensacola
Commodore Users of Pensacola
P O Box 3533
Debbie Johnson

Port Charlotte
Charlotte County Commodore Club
567 N Ellicott Cir
Lee Traux

Ruskin
C U S H
107 N Branch Rd, W
David Kinne

Titusville
Titusville Commodore Club, Inc
890 Alford St
Robert B Murray

GEORGIA

Albany
Albany Commodore Amateur Computer
P O Box 5461
Robert B McHugh

Athens
Athens Commodore Enthusiasts-(ACE)
130 St James Drive
Darrell Wesley

Decatur
Commodore Users' Group of Atlanta
633 Claimont Cir
Donald E Schwab

Savannah
SCUG C/O
103 Virginia Ave
Dale Reagon

HAWAII

Honolulu
Hawaiian Electric CUG
820 Ward Ave
Rodney Yim

IDAHO

Blackfoot
Blackfoot Users' Group
760 Janet St
Sam Wilcox

Pingree
The Blackfoot Users' Group
417 South 1200 West
Curtis Smith

ILLINOIS

Alton
East Side Computer Club
3103 Clay St
Dennis McGlasson

Argonne
Argonne Personal Computer Club
P O Box 337
Duane R Bradley

Aurora
Fox Valley 64 Users Group
1034 E Benton St
Frank Christensen

Bradley
Kankakee Hackers
200 E Broadway
Bill Brouillet

Canton
Canton Area Commodore Users Group
R R #1
Robert S Smolich

Chicago
Amoco Corp Microcomputer User Group
200 E Randolph—MC-1001
Richard Ryan

Darien
Suburban Commodore Users Group
P O Box 2201
Stan Niemiec

Galesburg
Knox Commodore Club
195 Olive St
Randy L Fox

Joliet
Com Ed Employee Commodore UG
1910 S Briggs
John McDowell

Lacon
Heart of IL Commodore Users Group
904 Fifth St
Robert Saville

Monmouth
Western IL Commodore Users Group
906 West 6th Ave
Robert Cokel

Murrayville
Jacksonville Area Commodore UG
4 Purcell St, P O Box 135
Greg Simpson

Springfield
Capitol City Commodore Computer
P O Box 2961
Mike Stout

Springfield
SPUG Computer Club
3166 Concord
Bill Eardley

INDIANA

Bloomington
Bloomington Commodore User's Group
800 E Atwater
Merrill J Allen

Evansville
Commodore Computer Club
P O Box 2332
Bruce Farley

Greenfield
C H U G
1322 Fairview Dr
Gregory A Chaney

Kokomo
Kokomo Commodore Computer Club
6016 Yale Court
Richard N LaMott

Lafayette
Commodore Owners of Lafayette
P O Box 5763
Ross Indelicato

Logansport
Logansport Commodore Club
2329 Myers Lane
Howard C Peoples

Terre Haute
Western Indiana Commodore Users
P O Box 1898

IOWA

Ames
COUGAR Commodore User Group
P O Box 2302
John Smith

Spencer
Syntax Errors Anonymous
1224-14th Avenue West
Russell S Oechslin

Waterloo
P E C C U G
333 Joy Dr
Al Sorensen

KANSAS

Atchison
Atchison Commodore Users Group
P O Box 396
Henry F Dove

Hays
High Plains Commodore User Group
1302 Western Plains Dr
Alan Clingingsmith

Hutchinson
Salt City Commodore Club
P O Box 2644
Wendell D Hinkson

Lawrence
Lawrence Commodore Users' Group
P O Box 2204
Joe Hewitt

KENTUCKY

Glasgow
Glasgow Commodore User Group, Inc
P O Box 154
Steve England

Lexington
Commodore UG of Central Kentucky
546 Halifax Dr
Virgil M Mills

Louisville
LUCKY
8601 McKenna Way
R D Bishop

Madisonville
Commodore UG of Madisonville CUGOM
534 Thompson Ave
Richard Byrd

LOUISIANA

Baton Rouge
Baton Rouge C-64 User Group
P O Box 1422
Byron Beverly, Jr

Lake Charles
Lake Charles Users' Group
5631 Lakelyn
Nelson Fontenot

Metairie
Sixty-Four'Um
4317 Stockton Street
Elizabeth S Hoffmann

Shreveport
Ark-La-Tex Commodore Computer Club
5515 Fairfax Ave
Bill Walker

Swartz
Northeast LA Commodore Users Group
P O Box 175
Beckie Walker

MAINE

Brunswick
Your Commodore User Group
175 Columbia Ave
Darlene English

Houlton
Southern Aroostook Commodore Users
P O Box 451
Robert Blanchette

MARYLAND

Baltimore
Randallstown Commodore Users' Group
3702 Durley Lane
Chris C Schildt

Hagerstown
Hagerstown User Group
23 Coventry Lane
Joseph F Rutkowski

Leonardtown
PAX/COM/64
240 Jefferson St
David J Hamblin

Reisterstown
CUM-BACC
P O Box 479
William J Kolodner

Riva
Annapolis Commodore User Group
P O Box Box 171
Don Chriscoe

Silver Spring
Vic Appreciators (VICAP)
10260 New Hampshire Avenue
Hugh S Pettis

Towson
BAYCUG
12 Walfred Court
Clewell Howell, Jr

MASSACHUSETTS

Fall River
M E M O
P O Box 3336
Kevin Cray

Rockland
Rockland Commodore User Group
98 Myrtle Street
Owen F Mahon

Westfield
Pioneer Valley Commodore Club
6 Laurel Terrace
Marvin S Yale

MICHIGAN

Bay City
Commodore Connection Club
400 18th Street
James Griffir

Clio
Northern Genesee County Commodore
P O Box 250
Garth J Sims

Dearborn
Ford Commodore Computer Club
P O Box 1201
Harry Tucker

Dollar Bay
Copper Country Computer Users Group
P O Box 196
Risto S Vuorinen

Dryden
ACO
4401 Pinegrove Dr
Arthur Nelson

E Lansing
Lansing Area Commodore Club
954 Trafalger
Jae Walker

Hudsonville
West Michigan Commodore User Group
3317 Van Buren
Ross A Ringerwole

Jackson
Jackson Commodore Computer Club
201 West Michigan
Steven M Bohne

Lennon
Commodore User Group of Durand, MI
11715 Park Dr
Marla Romine

Richmond
COMP
7514 Puttygut Rd
Brian Pringle

Southgate
Down River Commodore User Group
15331 Cameron
Nancy A Hanna

Ypsilanti
Washtenaw Commodore Users Group
4490 Oakengates Drive
Eugene McMurray

MINNESOTA

Rochester
Rochester Area Commodore User Group
2526 6th Ave NW
David Moertel

MISSISSIPPI

Biloxi
Coastline Commodore Computer Club
102 Monroe Circle
Norman R Cyr

MISSOURI

Ballwin
MDC-RCC Commodore Special Interest
566 Golf Wood Drive
Richard M Velten

Joplin
Joplin Commodore Computer User Group
424 S Florida Ave
Dale Connely

Morehouse
Delta Bootheel Users' Group
P O Box 16
Janet H Hinson

Warrensburg
Commodore User Group Warrensburg
P O Box 893
Buck Sommerkamp

MONTANA

Great Falls
Cascade County Commodore Users Group
P O Box 739
Gerald Spurbeck

Red Lodge
Grizzly Hackers
P O Box 1179
Les Dimich

Stevensville
Missoula Commodore Users
310 8th Street
John Verburg

NEBRASKA

Answorth
Sandhill Peeks & Pokes
237 N Park Street
Ray Stenka

Fremont
Pathfinder Commodore Users' Group
P O Box 683
Jack Quigley

Omaha
Greater Omaha Commodore Users Group
2932 Leawood Drive
Bob Quisenberry

NEW JERSEY

Cherry Hill
Amiga Users' Group of South Jersey
P O Box 3761
Jay Forman

Cherry Hill
S Jersey Amiga Users' Group
523 Society Hill
John Scott

Linwood
C-64 Users' Group of South Jersey
312 Beech Ave
Sam Levine

Matawan
Commodore User Group of Central NJ
112 Old Bridge Rd
David Habler

Roosevelt
Commodore Computer Collection Club
72 Pine Dr
Mel Friedman

Sewell
GC64 Commodore Users' Group
PO Box 292
Pete Achenbach

Westwood
Commodore 64 Beginners Club
680 Leigh Terrace
Tom Martin

NEW MEXICO

Las Cruces
Aviation & Computer Enthusiasts
1220 Birch Drive
Carl H Bogardus

Los Alamos
Los Alamos Commodore Users Group
3974-C Alabama St
Jack L Johnson

NEW YORK

Blue Mountain
Adirondack C-64 Users Group
P O Box 99
Jean Kerst

Brooklyn
Brooklyn Commodore User Group
1735 East 13th St, Apt 7-N
Malcolm J Gottesman

Glen Cove
Landing Users Group
17 Barlow Avenue
Don Bikoff

Glendale
Queens Commodore Users Group
75-11 64th Street
Bruce Behrend

Greene
NYTEC User Group
29 Juliand Street
Frank T Moorhead

Hammondsport
Finger Lakes Area Komputer Experts
86 W Lake Rd
Terri Lynne Narby

Holbrook
CLUB-64
1579 Coates Avenue
Michael McGrath

Levittown
Associated Commmodore Enthusiasts
37 Silver Lane
Chris Maselli

New Hampton
Commodore 64 U G of Orange County
P O Box 238
Stephen P Gerstl

New York
PC & Research Users Group
111 8th Ave, 11th FL
Lou Kairys

New York
Technology Not Tricks
619 W 114th Street
Tom Trocco

New York
The New York Times Personal Computer
229 West 43rd Street
Daniel Fersht

USER GROUPS

Oceanside
Commodore Long Island Club
2949 Roxbury Road
Eric P Miller

Oswego
Oswego Commodore User Group
208 Park Hall, Dept Technology
John R Boronkay

Tallman
Computer Club of Rockland
P O Box 233
Paul Franzel

Woodhaven
D - BUG
78 - 23 91st Avenue
Charles C Wagner

NEWFOUNDLAND CANADA

St John's
Pleasantville Commodore User Group
P O Box 2028
Greg Vincent

NORTH CAROLINA

Greensboro
Triad C-64 Users' Group
3302 Forsyth Dr
C W Blake

Hickory
Unifour Commodore Users Group
P O Box 9324
Janet M Morris

OHIO

Cincinnati
UCOM-64
340 Tangeman University Ctr #136
Bari Cruze

Columbus
Central Ohio Commodore User Group
P O Box 28229
Philip Lynch

Dayton
Page Manor Users' Group
P O Box 31744
Dennis R Paquette

Marion
Marion Ohio Commodore Users Group
775 Wolfinger Road
Richard Munro

New Philadelphia
CUG - Commodore 64 Users Group
702 Park Avenue N W
Jeff Eklund

North Canton
Canton/Akron/Massillon Area User Group
334 Fairview S E
Loren S Hines

Springfield
C H U G
P O Box 2238
Russell C Yeley

Toledo
Commodore Computer Club of Toledo
P O Box 8909
Jeff Hineline

OKLAHOMA

Bartlesville
Commodore Users of Bartlesville
1704 S Osage
Fred Mayes

OREGON

Eastside
Coos Computer Club
P O Box 4066
Patty Chard

Eugene
Lane County Commodore User Group
P O Box 11316
Maria Sayer

Oregon City
Northwest Amiga Group
P O Box 1140
Brian M Gagnier

PENNSYLVANIA

Bethlehem
ABC C-64 CHIPS
3159 Middletown Road, RD #3
Thomas Duff

Erie
North Coast Commodore User Group
P O Box 6117
Randall R Hodges

Feasterville
Lower Bucks User Group
P O Box 548
Robert G Stasche

New Kensington
AK-64 Users Group
1762 Fairmont Street
Alton E Glubish

Parksburg
Lighthouse Users Group
103 Peacemaker Dr
Matthew Henry

Philadelphia
SUB-64
P O Box 54208
Will Hines

Pittsburgh
Bettis Commodore User Group
592 Arbor Lane
Bill Campbell

Reading
Commodore User Group of Berks
300 Lackawanna St , Apt 9-F
Dave Procopio

Salona
Central PA User Group for Commodore
P O Box 1
Joseph W Coffman

West Chester
Main Line Commodore User Group
1046 General Allen Lane
Emil J Volcheck, Jr

SOUTH CAROLINA

Columbia
Commodore Computer Club of Columbia
318 Quincannon Drive
Buster White

Rock Hill
Rock Hill Commodore User Group
417 S Spruce St
Tim Johnson

SOUTH DAKOTA

Rapid City
Port-64
1705 Cruz Dr
Thomas S York

TENNESSEE

Estill Springs
Commodore Users' Club
P O Box 96
Marty Garner

Germantown
Memphis Amiga Group
PO Box 381462
Audrey B McCalla

Memphis
Memphis Commodore Users Group
P O Box 34095
Dwight Campbell

Springfield
Springfield Commodore User Group
Route #1, Box 166
Paul M S Bell

TEXAS

Amarillo
Tri State Commodore Users Group
P O Box 8971
Michael H Williamson

Austin
Commodore Computer Users of Texas
4001 N 1H35
Isaac Norris

Bay City
Bay City Commodore Users Group
2211 Gontier Drive
Glenn McCarthy

Bedford
Mid-Cities Commodore Club (MCCC)
P O Box 1578
William Raecke

Longview
Longview User Group
P O Box 9284
Joyce Pope

Texarkana
Ark-La-Tex Commodore Users
P O Box 6473
Stan Fierbaugh

Waco
Heart of Texas Home User Group
332-D Richland Drive
James Domengeaux

UTAH

Logan
Cache Valley Computer Club
467 N 200 W
Jody L Reese

VERMONT

S Burlington
Champlain Valley Commodore User Group
6 May Fair Street
Steve Lippert

VIRGINIA

Alexandria
Arlington VICtims
5521 Harvey Lane
Robin Franzel

Charlottesville
Piedmont Commodore Group
P O Box 5412
John A Stoner

Dale City
Dale City Commodore Users Group
P O Box 2265
Jack B Doyle

Haysi
Black Diamond Commodore Society
Rt 2, Box 628
Brian Hearl

Martinsville
Henry County Commodore Computer Club
P O Box 67
Ken Coffelt

Richmond
T R A C E The Richmond Area CE
2316 Lafayette Avenue
E M Rexrode

Virginia Beach
Tidewater Commodore Users Group
P O Box 61814
Mike Loding

WASHINGTON

Burley
Commodore Users of Puget Sound
P O Box 86
Kenneth A Blinn

Richland
Tri-City Commodore Computer Club
P O Box 1064
Jack Garvin

Seattle
University of Washington Commodore User Group
P O Box 75009
John Willcott

Spokane
Spokane Commodore Users Group
P O Box 1753
Steve Pretorius

Yakima
Central Washington Commodore User Group
P O Box 10937
Russell W Miller

WEST VIRGINIA

Bluefield
Bluefield User's Group C-64 & VIC
P O Box 1190
John M Knowles

Wheeling
C H U G Commodore Home Users Group
81 Lynwood Avenue
Alice Shipley

WISCONSIN

Green Bay
Comm Bay 64
Richard F Luxton

Greendale
M A C E
P O Box 183
Tim Pelzek

Janesville
Rock County Commodore Users Group
P O Box 1858
John Berg

Milwaukee
A O Smith Computer Club
P O Box 584
Ben Endejan

New Auburn
Chippewa Valley CUG
R 3
Norman C Hope

64 USERS ONLY/MAGIC MUSIC BOX

Continued from pg. 94

```
560 POKE 54273,NH(G):POKE 54272,NL(G)
    :POKE 54276,33'DHWN
570 FOR D=1 TO 77:NEXT D:POKE 54276,32
    :NEXT Q'GRXN
580 FOR X=1 TO 1234:NEXT:POKE 54273,0
    :POKE 54272,0:GOTO 130'HCJQ
600 REM THE PROGRAM ENDS WITH A SIREN
    / COLOR / PRINT ROUTINE'BSNN
620 FOR T=99 TO 181:PRINT"[RVS]
    THE END! ";'EIRI
625 POKE 53280,T:POKE 54273,250-T
    :POKE 54276,33'ECCP
630 POKE 53281,250-T:POKE 54276,32
    :NEXT:POKE 54273,0:FOR D=1 TO 1111
    :NEXT'JLKP
640 POKE 53281,6:POKE 53280,14
    :PRINT"[CLEAR,L. BLUE]":END'ESCL
700 REM DRAW PICTURE OF MAGIC MUSIC
    BOX'BBGJ
710 POKE 1384,79:POKE 1423,80
    :FOR T=1385 TO 1422:POKE T,119
    :NEXT'HHQM
715 FOR T=55656 TO 55695:POKE T,0
    :NEXT'FQDN
720 SP=LEN(M$):POKE 1344+SP,78
    :POKE 1344+SP+39,32
    :POKE 1344+SP+38,32'KPCS
725 POKE 55616+SP,0:POKE 1344+SP-1,
    32'FUHP
730 POKE 1424,116:POKE 1463,106
    :POKE 55696,0:POKE 55735,0'EIYM
740 POKE 1464,76:POKE 1503,122
    :FOR T=1465 TO 1502:POKE T,111
    :NEXT'HIVP
745 FOR T=55736 TO 55775:POKE T,0
    :NEXT'FQBQ
750 RETURN'BAQG
800 REM MAKE THE MAGIC MUSIC BOX
    PROCESS THE MELODY'BLJN
810 G=RND(1)*38:C=RND(1)*16
    :J=(RND(1)*11)+48:I=SP-1
    :N=RND(1)*10'PLKU
820 POKE 1425+G,J:POKE 55697+G,C
    :POKE,1344+I,86:POKE 55616+I,
    C'IJKQ
830 POKE 54273,NH(N):POKE 54272,NL(N)
    :POKE 54276,17:FOR D=1 TO 3
    :POKE 54276,16'HUAS
840 RETURN'BAQG
900 REM OPEN THE MAGIC MUSIC BOX AND
    LET THE MELODY OUT'BNYO
910 FOR T=55736 TO 55775:POKE T,0:NEXT
    :POKE 1471,32:POKE 1472,32'HHVO
920 POKE 1511,77:POKE 55783,0'CPYJ
930 PRINT"[HOME,DOWN12,RIGHT10]
    AND IT COMES OUT HERE!"'BADO
940 FOR D=1 TO 1111:NEXT'EHDK
950 RETURN'BAQI
```

END

JIFFIES/THE CLOCK

Continued from pg. 117

```
914 DATA 162,1,173,11,220,41,128,
    240'BCRN
916 DATA 2,162,16,138,153,26,4,
    200'BAAP
918 DATA 169,13,153,26,4,200,169,
    32'BBJR
920 DATA 153,26,4,200,173,11,220,
    170'BCQK
922 DATA 41,16,32,1,192,173,10,
    220'BASM
924 DATA 32,0,192,173,9,220,32,0'BXDN
926 DATA 192,173,8,220,41,15,9,48'BYQQ
928 DATA 153,26,4,169,1,160,13,
    153'BAFS
930 DATA 26,216,136,208,250,76,49,
    234'BDUL
932 DATA 120,169,27,141,20,3,169,
    192'BCKN
934 DATA 141,21,3,88,96,256'BSHO
950 POKE SA+106,FN L(SA+27)'ENDN
955 POKE SA+111,FN H(SA+27)'ENUS
960 READ I,P: IF I=-1 THEN 985'FJCO
965 POKE SA+I,FN L(SA+P)'EKAT
966 POKE SA+I+1,FN H(SA+P)'FLPU
970 GOTO 960'BDOL
980 DATA 30,20,67,1,73,0,79,0,-1,
    0'BAHQ
985 POKE U,PEEK(U)OR 1:RETURN'EHPU
```

END

Falcon and the Enemy

Continued from pg. 71

```
     [SHFT -,L. RED]AWAITS[CYAN]
     [SHFT -]"+R$+"[CMDR Z,SHFT *7,
     CMDR X]"'HMVU
1045 PRINT RI$RP$'BGEE
1050 PRINT"[HOME,DOWN,RVS,RED]
      YOU HAVE FAILED! [RVOFF]"
     :GOSUB 3000:PRINT"[DOWN,GREEN,
     SPACE3]THE RESULTS OF"'DGJK
1051 PRINT" YOUR EFFORTS LIE"
     :PRINT" TO THE RIGHT."'CBHI
1052 PRINT" [PURPLE,CMDR @15,RVS,
     CMDR *,RVOFF]":PRINT" [CMDR T15,
     SHFT POUND]":GOSUB 3000'DGWA
1053 PRINT"[DOWN,RED,SPACE5,RVS]
      HOWEVER [RVOFF]":GOSUB 3000
     :PRINT"[SPACE3,GREEN]
     THE MOTHER SHIP"'DGAM
1054 PRINT" WAS ABLE TO FIGHT"
     :PRINT" OFF THE ATTACK."'CBDM
1055 GOSUB 3000:PRINT"[DOWN,SPACE3]
     YOU STILL HAVE":PRINT" A
     CHANCE."'DGFM
1056 GOSUB 3000:PRINT"[DOWN,SPACE3]
     IF YOU WISH":PRINT" ANOTHER
     MISSION?"'DGFP
1057 PRINT"[DOWN,PURPLE,RVS,^19]"
     :PRINT"[RVS,^2,RVOFF,GREEN]
     PRESS SPACE BAR[PURPLE,RVS,^2]
     "'CBPV
1058 PRINT"[RVS,^19,HOME,GREEN]"'BAHN
1060 GET A$:IF A$=""THEN 1060'EJED
1061 IF A$=" "THEN PRINT"[CLEAR,GREEN]
     ":RESTORE:GOTO 10'GGHF
1999 END'BACU
2000 POKE 49989,234:POKE 49994,49
     :SYS 49987:S=54272'EGGD
2005 FOR L=S TO S+24:POKE L,0:NEXT
     :POKE V+21,0:PRINT"[CLEAR,GREEN]"
     :RETURN'KSJJ
3000 FOR T=0 TO 950:NEXT T:RETURN'FIIY
4000 PRINT"[CLEAR]";'BBDW
4010 V=53248:REM *******************
     **'CEOD
4011 POKE 2040,192:POKE V,24
     :POKE V+1,142:POKE V+39,7:'GCYH
4012 POKE 2041,193:POKE V+2,248
     :POKE V+3,142:POKE V+40,7:'HEPJ
4013 POKE 2042,194:POKE V+4,136
     :POKE V+5,50:POKE V+41,7:'HDSK
4014 POKE 2043,195:POKE V+6,136
     :POKE V+7,229:POKE V+42,7:'HEGL
4015 POKE V+43,5:POKE V+28,192
     :POKE V+37,2:POKE V+38,7'IADM
4016 POKE 2046,200:POKE 2047,199
     :POKE V+46,1:'EYUK
4020 Z$="[RIGHT31]":Z1$="[RVS,RED,
     SPACE9,RVOFF]"'CGVG
4021 Z2$="[RVS,RED] [RVOFF,SPACE7,RVS]
      [RVOFF]":Z3$="[RVS,RED] [PURPLE,
     ^7,RED] [RVOFF]":Z4$="[RVS,RED]
     [PURPLE,^,GREEN,RVOFF]TIME![RVS,
```

```
      PURPLE,^,RED] [RVOFF]"'DLLP
4022 Z5$="[RVS,RED] [PURPLE,^,GREEN,
      RVOFF]SCORE[RVS,PURPLE,^,RED]
      [RVOFF]":Z6$="[RVS,RED] [PURPLE,
      ^,GREEN,RVOFF]YOUR![RVS,PURPLE,^,
      RED] [RVOFF]"'CHTN
4023 BL$="[HOME,GREEN,DOWN8,RIGHT33,
      SPACE5,LEFT5]"'BDSK
4024 G$="[HOME,GREEN,RIGHT33,DOWN2,
      SPACE4,LEFT4]":XX=49625'CKCM
4025 AL$="[HOME,DOWN15,RIGHT32]"'BDVJ
4026 I$="[RVS]":O$="[CYAN]A[SPACE6,
      DOWN,LEFT7] L[SPACE5,DOWN,LEFT7,
      SPACE2]I[SPACE4]"'CFPT
4027 O$=O$+"[DOWN,LEFT7,SPACE3]V
      [SPACE3,DOWN,LEFT7,SPACE4]E
      [SPACE2,DOWN,LEFT7,SPACE5]!
      [RVOFF]"'CEBX
4050 PRINT"[CLEAR]";Z$;Z1$;Z$;Z2$;Z$;
      Z2$;Z$;Z2$;Z$;Z3$;Z$;Z4$;Z$;Z3$;
      Z$;Z1$;Z$;'BKQQ
4051 PRINT Z2$;Z$;Z2$;Z$;Z2$;Z$;Z3$;
      Z$;Z5$;Z$;Z3$;Z$;Z1$;Z$;Z2$;Z$;
      Z2$;Z$;'BNVR
4052 PRINT Z2$;Z$;Z2$;Z$;Z2$;Z$;Z2$;
      Z$;Z3$;Z$;Z6$;Z$;Z3$;Z$;"[RVS,
      RED,SPACE8,RVOFF,HOME]";'BAQR
4053 POKE 2023,160:PRINT BL$"[DOWN,
      LEFT,RED,SHFT *7]";:POKE 56295,2
      :'DWCQ
4100 FOR Q=12288 TO 12522:POKE Q,0
      :NEXT Q'FRED
4101 FOR Q=12297 TO 12339 STEP 3
      :READ A: POKE Q,A:NEXT Q'HUSG
4102 FOR Q=12363 TO 12405 STEP 3
      :READ A: POKE Q,A:NEXT Q'HUGH
4103 FOR Q=12416 TO 12436:READ A
      :POKE Q,A:NEXT Q'GTUH
4104 FOR Q=12522 TO 12542:READ A
      :POKE Q,A:NEXT Q'GTQI
4105 FOR Q=12544 TO 12606:READ A
      :POKE Q,A:NEXT Q'GTVJ
4106 FOR Q=12608 TO 12670:READ A
      :POKE Q,A:NEXT Q'GTXK
4107 FOR Q=12672 TO 12734:POKE Q,0
      :NEXT Q'FRGK
4108 FOR Q=12694 TO 12712 STEP 3
      :READ A: POKE Q,A:NEXT Q'HUON
4109 FOR Q=12736 TO 12798:READ A
      : POKE Q,A:NEXT Q'GTLN
4110 FOR Q=12800 TO 12862:READ A
      : POKE Q,A:NEXT Q'GTUF
4111 FOR X=55296 TO 55416 STEP 40
      :FOR A=0 TO 30:POKE X+A,1:NEXT A
      :NEXT X'LCDL
4112 FOR X=56136 TO 56256 STEP 40
      :FOR A=0 TO 30:POKE X+A,1:NEXT A
      :NEXT X'LCAM
4113 FOR X=55456 TO 56096 STEP 40
      :FOR A=0 TO 5:POKE X+A,1
      :POKE X+A+25,1:NEXT:NEXT'OGMQ
4114 FOR Y=55462 TO 56102 STEP 40
      :FOR A=0 TO 18:POKE Y+A,12:NEXT A
      :NEXT Y'LDAO
4115 FOR Z=55668 TO 55908 STEP 40
      :FOR A=0 TO 6:POKE Z+A,11:NEXT A
      :NEXT Z'LCWP
4200 S=54272:FOR L=S TO S+24:POKE L,0
      :NEXT L:POKE S+24,15'JAXJ
4205 REM ******** VOICE #1
      ***********'BBSI
4210 POKE S+5,10:POKE S+6,242
      :POKE S+1,255:'GUFH
4215 REM ******** VOICE #2
      ***********'BBTJ
4220 POKE S+12,1:POKE S+13,242
      :POKE S+8,25:POKE S+9,9
      :POKE S+10,55'KGUN
4225 REM ******** VOICE #3
      ***********'BBUK
4230 POKE S+19,31:POKE S+20,241
      :POKE S+15,5:POKE S+18,129'IDVM
4500 RETURN'BAQB
4600 REM ********  M - L
      ************'BXGG
4601 DATA 141,113,5,141,208,4,141'BXDH
4602 DATA 90,4,141,79,5,141,179'BVDI
4603 DATA 4,141,144,4,141,125,5'BVHJ
4604 DATA 141,253,5,141,193,6,141'BXNK
4605 DATA 106,6,141,132,7,141,7'BVKL
4606 DATA 7,141,201,7,141,178,6'BVQM
4607 DATA 141,211,6,141,232,5,169'BXLN
4608 DATA 192,141,227,193,141,221,
      193'BCNP
4609 DATA 169,0,141,226,193,169,
      67'BYCQ
4610 DATA 141,220,193,96,141,71,5'BXSH
4611 DATA 141,126,4,141,9,4,141'BVJI
4612 DATA 39,5,141,100,4,141,27'BVIJ
4613 DATA 4,141,87,5,141,241,5'BUPK
4614 DATA 141,235,6,141,188,6,141'BXSL
4615 DATA 213,7,141,87,7,141,87'BVCM
4616 DATA 7,141,0,7,141,249,6'BTQN
4617 DATA 141,229,5,169,192,141,
      221'BAOP
4618 DATA 193,141,227,193,169,134,
      141'BCUQ
4619 DATA 220,193,169,67,141,226,
      193'BBYR
4620 DATA 96,141,29,5,141,3,4'BTTI
4621 DATA 141,3,4,141,215,4,141'BVCJ
4622 DATA 21,4,141,200,5,141,49'BVEK
4623 DATA 5,141,242,5,141,21,7'BUHL
4624 DATA 141,14,7,141,14,7,141'BVHM
4625 DATA 207,7,141,62,6,141,78'BVWN
4626 DATA 7,141,21,6,141,225,5'BUJO
4627 DATA 169,192,141,221,193,141,
      227'BCQQ
4628 DATA 193,169,201,141,220,193,
      169'BCSR
4629 DATA 134,141,226,193,96,141,
      243'BBQS
4630 DATA 4,141,198,5,141,158,5'BVXJ
4631 DATA 141,135,4,141,135,4,141'BXFK
4632 DATA 161,5,141,11,5,141,243'BWEL
4633 DATA 5,141,25,6,141,96,7'BTVM
4634 DATA 141,64,6,141,63,6,141'BVON
4635 DATA 141,6,141,195,7,141,59'BWXO
4636 DATA 6,141,237,5,169,192,141'BXYP
```

```
4637 DATA 227,193,169,193,141,221,
     193'BCYR
4638 DATA 169,201,141,226,193,169,
     14'BBSS
4639 DATA 141,220,193,96,141,201,
     4'BYET
4640 DATA 141,157,5,141,77,5,141'BWSK
4641 DATA 15,4,141,160,5,141,122'BWCL
4642 DATA 5,141,229,4,141,244,5'BVOM
4643 DATA 141,67,6,141,219,7,141'BWTN
4644 DATA 145,6,141,103,6,141,220'BXDO
4645 DATA 6,141,22,6,141,97,6'BTTP
4646 DATA 141,236,5,169,193,141,
     227'BATR
4647 DATA 193,141,221,193,169,14,
     141'BBNS
4648 DATA 226,193,169,81,141,220,
     193'BBUT
4649 DATA 96,141,197,5,141,116,5'BWAT
4650 DATA 141,252,4,141,159,5,141'BXML
4651 DATA 81,5,141,44,5,141,201'BVHM
4652 DATA 5,141,246,5,141,109,6'BVPN
4653 DATA 141,24,6,141,226,6,141'BWJO
4654 DATA 143,6,141,43,7,141,61'BVMP
4655 DATA 6,141,135,6,141,235,5'BVNQ
4656 DATA 169,193,141,221,193,141,
     227'BCRS
4657 DATA 193,169,81,141,226,193,
     169'BBHT
4658 DATA 148,141,220,193,96,141,
     155'BBRU
4659 DATA 5,141,34,5,141,171,4'BUJU
4660 DATA 141,119,5,141,2,5,141'BVGM
4661 DATA 222,4,141,163,5,141,249'BXNN
4662 DATA 5,141,151,6,141,65,6'BUOO
4663 DATA 141,51,7,141,183,6,141'BWMP
4664 DATA 122,7,141,100,6,141,173'BXDQ
4665 DATA 6,141,234,5,169,193,141'BXWR
4666 DATA 227,193,169,192,141,221,
     193'BCXT
4667 DATA 169,0,141,220,193,169,
     148'BATU
4668 DATA 141,226,193,96,169,46,
     32'BYDV
4669 DATA 148,193,96,169,32,32,81'BXHW
4670 DATA 193,96,173,254,3,208,15'BXAO
4671 DATA 173,53,3,141,14,208,173'BXPO
4672 DATA 54,3,141,15,208,141,254'BXMP
4673 DATA 3,165,2,74,144,6,206'BUSQ
4674 DATA 15,208,74,16,6,74,144'BVAR
4675 DATA 3,238,15,208,74,144,5'BVWS
4676 DATA 206,14,208,176,6,74,144'BXYT
4677 DATA 3,238,14,208,173,14,208'BXSU
4678 DATA 201,35,176,5,238,14,208'BXTV
4679 DATA 208,7,201,237,144,3,206'BXNW
4680 DATA 14,208,173,15,208,201,
     50'BYDP
4681 DATA 176,5,238,15,208,208,7'BWDP
4682 DATA 201,229,144,3,206,15,
     208'BYFR
4683 DATA 96,173,0,220,41,31,73'BVOR
4684 DATA 31,208,3,76,207,194,74'BWCS
4685 DATA 144,6,206,1,208,206,3'BVKT
4686 DATA 208,74,144,6,238,1,208'BWXU
4687 DATA 238,3,208,74,144,6,206'BWXV
4688 DATA 4,208,206,6,208,74,144'BWVW
4689 DATA 6,238,4,208,238,6,208'BVCX
4690 DATA 74,144,43,173,52,3,208'BWVP
4691 DATA 38,169,65,141,11,212,
     173'BYPR
4692 DATA 4,208,141,8,208,173,1'BVPR
4693 DATA 208,141,9,208,169,196,
     141'BAYT
4694 DATA 252,7,169,21,141,52,3'BVQT
4695 DATA 141,252,3,169,16,13,21'BWLU
4696 DATA 208,141,21,208,173,4,
     208'BYHW
4697 DATA 201,34,176,8,238,4,208'BWWW
4698 DATA 238,6,208,208,10,201,
     238'BYJY
4699 DATA 144,6,206,4,208,206,6'BVQY
4700 DATA 208,173,1,208,201,59,
     176'BYQI
4701 DATA 8,238,1,208,238,3,208'BVXI
4702 DATA 208,10,201,221,144,6,
     206'BYVK
4703 DATA 1,208,206,3,208,206,252'BXKK
4704 DATA 3,208,35,169,20,141,252'BXPL
4705 DATA 3,173,252,7,201,198,240'BXSM
4706 DATA 5,238,252,7,208,18,169'BWHN
4707 DATA 0,141,52,3,169,239,45'BVUO
4708 DATA 21,208,141,21,208,169,
     64'BYLQ
4709 DATA 141,11,212,206,253,3,
     208'BYXR
4710 DATA 13,169,191,45,21,208,
     141'BYMJ
4711 DATA 21,208,169,128,141,4,
     212'BYIK
4712 DATA 173,30,208,240,50,201,
     144'BAXL
4713 DATA 208,40,45,21,208,168,
     173'BYPM
4714 DATA 14,208,141,12,208,173,
     15'BYEN
4715 DATA 208,141,13,208,152,9,79'BXYN
4716 DATA 141,21,208,169,129,141,
     4'BYKP
4717 DATA 212,206,255,2,169,0,141'BXKP
4718 DATA 254,3,169,10,141,253,3'BWNQ
4719 DATA 172,30,208,172,30,208,
     76'BYMS
4720 DATA 49,234,120,169,194,141,
     21'BAPK
4721 DATA 3,169,63,141,20,3,88'BUWK
4722 DATA 96,234,234,234,234,234,
     234'BBSM
5000 REM ********   LEFT
     ************'BYCC
5001 DATA 128,192,224,240,248,252,254,
     255,254,252,248,240,224,192,
     128'BJTJ
5002 REM ********   RIGHT
     ************'BAJE
5003 DATA 1,3,7,15,31,63,127,255,127,
     63,31,15,7,3,1'BQLI
5004 REM ********    TOP
     ************'BXBF
5005 DATA 7,255,224,3,255,192,1,255,
```

Falcon and the Enemy

```
        128,0,255,0,0,126,0,0,60,0,0,24,
        0'BJCN
5006 REM ********  BOTTOM
        ************'BBVI
5007 DATA 0,24,0,0,60,0,0,126,0,0,255,
        0,1,255,128,3,255,192,7,255,
        224'BJCP
5008 REM ********  SHOT-L
        ************'BBQK
5009 DATA 0,24,0,0,24,0,0,24,0,0,24,0,
        0,24,0,0,24,0,0,24,0,0,0,0,0,
        0'BKHR
5010 DATA 0,0,0,255,0,255,0,0,0,0,0,
        0,0,0,0,24,0,0,24,0,0,24,0,0,24,
        0,0'BNXK
5011 DATA 24,0,0,24,0,0,24,0'BSQC
5012 REM ********  SHOT-M
        ************'BBRF
5013 DATA 0,0,0,0,0,0,0,0,0,0,0,0,
        24,0,0,24,0,0,24,0,0,24,0,0,
        24'BGNL
5014 DATA 0,0,24,0,7,231,224,0,24,0,0,
        24,0,0,24,0,0,24,0,0,24,0,0,
        24'BITN
5015 DATA 0,0,0,0,0,0,0,0,0,0,0,0,
        0'BADI
5016 REM ********  SHOT-S
        ************'BBXJ
5017 DATA 24,24,24,255,24,24,24'BVNJ
5018 REM ********  SHIP
        ************'BYLL
5019 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
        0,0,0,0,0,0,0,0,0,0,0,170,0,
        10'BIAS
5020 DATA 170,160,165,85,90,42,85,168,
        2,170,128,0,0,0,0,0,0,0,0,0'BFNJ
5021 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
        0'BELF
5022 REM ********  BLAST
        ************'BABG
5023 DATA 0,0,0,20,0,20,21,65,84,5,85,
        80,5,245,80,23,255,84,85,255,213,
        21'BNQO
5024 DATA 127,84,5,95,208,23,255,84,
        85,245,85,21,253,84,5,125,80,21,
        255'BLRP
5025 DATA 212,87,255,85,23,255,84,7,
        87,208,7,85,80,21,65,84,20,0,20,
        0,0,0'BNMQ
```

END

AMIGA UPDATE

Continued from pg. 111

new date and time, show you once more what date it is, and load the Workbench. You may have noticed that ECHO " " was used. This merely separates the text, similar to PRINT alone in BASIC. However, ECHO by itself is not acceptable.

Yesterday, Today, Future

Even though you have entered the correct date and time, AmigaDOS will print YESTERDAY, TODAY or FUTURE at times. If you save a program on August 7, 1986, and check the disk directory that same day, you'll find "TODAY," since you read the directory the same day you saved it. If you decide to enter the next date, August 8, and check the directory, you'll find the word "YESTERDAY." And finally, if you enter a date such as January 30, 1986, and list the directory, "FUTURE" will be displayed. Any other date after "August 8" will have the directory show the original date saved, "August 7."

As you can see, your Amiga has more intelligence than you think. Now that you have a "date-setting" program, you'll never have programs saved under the date January 1, 2001, again! C

GLITCH FIX

Magazine Entry Program – 128

There has been a bug in the 128 version of the Magazine Entry Program which will generate a NO CHECKSUM error on any line which contains only a colon. The correction to the program is very simple: In line 110, change the value 59314 to 59311. In line 1040 change the 6th data item (Ø9) to a 6 (Ø6).

```
11Ø IF T<>59311 THEN PRINT
    :PRINT"MISTAKE IN DATA --> CHECK
    DATA STATEMENTS":END

1Ø4Ø DATA B1,3D,DØ,FB,CØ,Ø6,1Ø,Ø3
```

E.S.P. – January 1987 Issue

Two lines in the E.S.P. program were printed incorrectly in the January issue of the magazine. Here are the correct lines and checksums:

```
4ØØ POKE S+24,15:CØ=3:CS=Ø:WS=Ø
    :DECK=25:PRINT"[CLEAR,DOWN5,
    SPACE2,WHITE,SHFT R]IGHT"SPC(24)"
    [SHFT W]RONG"'IEYP

43Ø JOY=NOT PEEK(5632Ø)AND 15
    :FIRE=(PEEK(5632Ø)AND 16)/16
    :IF FIRE=Ø THEN 47Ø'LOJR
```


ADVERTISERS INDEX

*No Reader Service Number given at advertiser's request.